18e Conférence en Recherche d'Information et Applications
16e Rencontres Jeunes Chercheurs en RI
30e Conférence sur le Traitement Automatique des Langues Naturelles
25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues
(CORIA-TALN) [1]

Actes de CORIA-TALN 2023.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN),
volume 1 : travaux de recherche originaux – articles longs

Christophe Servan, Anne Vilnat (Éds.)

Paris, France, 5 au 9 juin 2023

1. https://coria-taln-2023.sciencesconf.org/

Avec le soutien de

Préface

Organisée conjointement par les laboratoires franciliens sous l'égide de l'Association francophone de Recherche d'Information et Applications (ARIA) et l'Association pour le Traitement Automatique des Langues (ATALA), la conférence CORIA-TALN-RJCRI-RECITAL 2023 regroupe :

— la 18ème Conférence en Recherche d'Information et Applications (CORIA)
— la 30ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN) ;

ainsi que les deux conférences associées, destinées aux jeunes chercheuses et chercheurs :

— Les 16ème Rencontres Jeunes Chercheurs en RI (RJCRI)
— la 25ème Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL)

La conférence TALN (Traitement Automatique des Langues Naturelles) est un rendez-vous annuel qui offre, depuis 1994, le plus important forum d'échange international francophone aux acteurs universitaires et industriels des technologies de la langue. Cet événement, qui accueille habituellement près de 250 participants, couvre toutes les avancées récentes en matière de communication écrite et parlée et de traitement informatique de la langue notamment la recherche et l'extraction d'information, la fouille de textes, le dialogue homme-machine, la fouille d'opinions, la traduction automatique, les systèmes de questions-réponses, le résumé automatique...

Cette année, ont été soumis 51 articles longs et 12 articles courts pour la conférence principale, dont respectivement 29 ont été acceptés pour une présentation orale (dont 2 prises de position) et 9 pour une présentation sous forme de posters. 19 présentations courtes, sous forme de posters, d'articles déjà publiés lors de conférences internationales complètent le programme de la conférence, ainsi que des démonstrations et des présentations de projets en cours. L'alternance de sessions communes entre TALN, CORIA et RJC et de sessions plus spécifiques devraient permettre de susciter des échanges fructueux.

En complément de la conférence principale, se tiennent les ateliers "Défi Fouille de Texte" (DEFT), "Atelier sur l'analyse et la recherche de textes scientifiques" (ARTS), "Humain ou pas humain ? : les nouveaux défis pour les humains" (hOUPSh) et le tutoriel "Apprentissage Profond pour le TAL français pour les debutants" (TutoriAL). Ces ateliers et tutoriel illustrent à la fois des tendances nouvelles présentes dans la communauté et des activités récurrentes.

Un grand merci à toutes celles et tous ceux qui ont soumis leurs travaux, ainsi qu'aux membres du comité de programme et aux relectrices et relecteurs pour le travail qu'ils ont accompli. Ce sont eux qui font vivre la conférence. Merci au comité d'organisation réparti sur la région parisienne, et aux sponsors qui nous ont permis d'organiser cet événement.

Christophe Servan et Anne Vilnat, co-présidents de TALN

Comités

Comité de programme

Présidents

— Christophe SERVAN
— Anne VILNAT

Membres

— Rachel BAWDEN
— Caroline BRUN
— Marie CANDITO
— Rémi CARDON
— Pascal DENIS
— Yannick ESTEVE
— Benoît FAVRE
— Amel FRAISSE
— Thomas GERALD
— Natalia GRABAR
— Lydia-Mai HO-DAC
— José MORENO
— Vassilina NIKOULINA
— Yannick PARMENTIER
— Sylvain POGODALLA
— Solène QUINIOU
— Didier SCHWAB
— Iris TARAVELLA-ESHKOL

Comité d'organisation

— Marie CANDITO
— Thomas GERALD
— José MORENO
— Benjamin PIWOWARSKI
— Christophe SERVAN
— Laure SOULIER
— Anne VILNAT

Table des matières

Étude de méthodes d'augmentation de données pour la reconnaissance d'entités nommées en astrophysique **1**
Atilla Kaan Alkan, Cyril Grouin, Pierre Zweigenbaum

Towards a Robust Detection of Language Model-Generated Text : Is ChatGPT that easy to detect ? **14**
Wissam Antoun, Virginie Mouilleron, Benoît Sagot, Djamé Seddah

Cross-lingual Strategies for Low-resource Language Modeling : A Study on Five Indic Dialects **28**
Niyati Bafna, Cristina España-Bonet, Josef Van Genabith, Benoît Sagot, Rachel Bawden

Pauzee : Prédiction des pauses dans la lecture d'un texte **43**
Marion Baranes, Karl Hayek, Romain Hennequin, Elena V. Epure

Reconnaissance de défigements dans des tweets en français par des mesures de similarité sur des alignements textuels **56**
Julien Bezançon, Gaël Lejeune

Tri-apprentissage génératif : génération de données pour de la reconnaissance d'entitées nommées semi-supervisé **68**
Hugo Boulanger, Thomas Lavergne, Sophie Rosset

Évaluation d'un générateur automatique de reformulations médicales **80**
Ioana Buhnila, Amalia Todirascu

Étude comparative des plongements lexicaux pour l'extraction d'entités nommées en français **94**
Danrun Cao, Nicolat Béchet, Pierre-François Marteau

Honey, Tell Me What's Wrong, Explicabilité Globale des Modèles de TAL par la Génération Coopérative **105**
Antoine Chaffin, Julien Delaunay

Extraction de relations sémantiques et modèles de langue : pour une relation à double sens **123**
Olivier Ferret

Géométrie de l'auto-attention en classification : quand la géométrie remplace l'attention **137**
Loïc Fosse, Duc Hau Nguyen, Pascale Sébillot, Guillaume Gravier

Un traitement hybride du vague textuel : du système expert VAGO à son clone neuronal **151**
Benjamin Icard, Vincent Claveau, Ghislain Atemezing, Paul Egré

Uniformité de la densité informationnelle : le cas du redoublement du sujet **164**
Yiming Liang, Pascal Amsili, Heather Burnett

Augmentation des modèles de langage français par graphes de connaissances pour la reconnaissance des entités biomédicales **177**
Aidan Mannion, Schwab Didier, Lorraine Goeuriot, Thierry Chevalier

Annotation d'entités cliniques en utilisant les Larges Modèles de Langue **190**
Simon Meoni, Théo Ryffel, Eric De La Clergerie

Classification de tweets en situation d'urgence pour la gestion de crises **204**
Romain Meunier, Véronique Moriceau, Patricia Stolf, Farah Benamara, Leila Moudjari, Alda Mari

Outiller l'occitan : nouvelles ressources et lemmatisation **217**
Aleksandra Miletić

Stratégies d'apprentissage actif pour la reconnaissance d'entités nommées en français **232**
Marco Naguib, Aurélie Névéol, Xavier Tannier

Détecter une erreur dans les phrases coordonnées au sein des rédactions universitaires **248**
Laura Noreskal, Iris Eshkol-Taravella, Marianne Desmets

Production automatique de gloses interlinéaires à travers un modèle probabiliste exploitant des alignements **262**
Shu Okabe, François Yvon

Intégration de connaissances structurées par synthèse de texte spécialisé **275**
Guilhem Piat, Ellington Kirby, Julien Tourille, Nasredine Semmar, Alexandre Allauzen, Hassane Essafi

DACCORD : un jeu de données pour la Détection Automatique d'énonCés COntRaDictoires en français **285**
Maximos Skandalis, Richard Moot, Simon Robillard

Exploitation de plongements de graphes pour l'extraction de relations biomédicales **298**
Anfu Tang, Robert Bossy, Louise Deléger, Claire Nédellec, Pierre Zweigenbaum

Derrière les plongements de relations **311**
Hugo Thomas, Guillaume Gravier, Pascale Sébillot

CamemBERT-bio : Un modèle de langue français savoureux et meilleur pour la santé **323**
Rian Touchent, Laurent Romary, Eric De La Clergerie

Protocole d'annotation multi-label pour une nouvelle approche à la génération de réponse socio-émotionnelle orientée-tâche **335**
Lorraine Vanel, Alya Yacoubi, Chloe Clavel

Exploring Data-Centric Strategies for French Patent Classification : A Baseline and Comparisons **349**
You Zuo, Benoît Sagot, Kim Gerdes, Houda Mouzoun, Samir Ghamri Doudane

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Étude de méthodes d'augmentation de données pour la reconnaissance d'entités nommées en astrophysique

Atilla Kaan Alkan[1, 2] Cyril Grouin[1] Pierre Zweigenbaum[1]
(1) Université Paris-Saclay, CNRS, Laboratoire interdisciplinaire des sciences du numérique, 91405, Orsay, France.
(2) IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France.
{atilla.alkan, cyril.grouin, pz}@lisn.upsaclay.fr

RÉSUMÉ

Dans cet article nous étudions l'intérêt de l'augmentation de données pour le repérage d'entités nommées en domaine de spécialité : l'astrophysique. Pour cela, nous comparons trois méthodes d'augmentation en utilisant deux récents corpus annotés du domaine : DEAL et TDAC, tous deux en anglais. Nous avons générés les données artificielles en utilisant des méthodes à base de règles et à base de modèles de langue. Les données ont ensuite été ajoutées de manière itérative pour affiner un système de détection d'entités. Les résultats permettent de constater un effet de seuil : ajouter des données artificielles au-delà d'une certaine quantité ne présente plus d'intérêt et peut dégrader la F-mesure. Sur les deux corpus, le seuil varie selon la méthode employée, et en fonction du modèle de langue utilisé. Cette étude met également en évidence que l'augmentation de données est plus efficace sur de petits corpus, ce qui est cohérent avec d'autres études antérieures. En effet, nos expériences montrent qu'il est possible d'améliorer de 1 point la F-mesure sur le corpus DEAL, et jusqu'à 2 points sur le corpus TDAC.

ABSTRACT

Investigating Data Augmentation Methods for Astrophysical Named Entity Recognition.

In this paper, we investigate the effectiveness of data augmentation for named entity recognition in astrophysics. To this end, we compare three augmentation methods using two recent annotated corpora in the domain : DEAL and TDAC, both in English. We generated artificial data using rule-based and language model-based approaches. The data was then iteratively added to finetune an entity detection system. The results show a threshold effect : adding artificial data beyond a specific quantity is no longer beneficial and can decrease F-measure. The threshold varies for each method and depends on the language model employed. This study also highlights that data augmentation is more effective for small corpora, consistent with previous studies. Indeed, our experiments demonstrate the potential to improve the F-measure by 1 point in the DEAL corpus and up to 2 points in the TDAC corpus.

MOTS-CLÉS : Repérage d'entités nommées, Augmentation de données, Annotation, Astrophysique.

KEYWORDS: Named Entity Recognition, Data Augmentation, Annotation, Astrophysics.

1 Introduction

Tâche de base en Traitement Automatique des Langues (TAL), la reconnaissance d'entités nommées (REN) consiste à repérer des mentions d'entités dans un texte afin de les catégoriser dans des classes

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 1–13.

pré-définies. Depuis son introduction en 1996 (Grishman & Sundheim, 1996), cette tâche s'est avérée utile pour la recherche d'information (Banerjee *et al.*, 2019) ou encore pour la constitution de systèmes de questions-réponses (Mollá Aliod *et al.*, 2006). Cependant, l'entraînement des systèmes de repérage d'entités nommées par apprentissage requièrent une quantité significative de données annotées, qui ne sont pas toujours disponibles pour certaines langues, peu dotées en ressources linguistiques, ou encore en domaine de spécialité.

Dans cet article, nous étudions l'intérêt de l'augmentation de données pour le repérage d'entités nommées, dans un domaine de spécialité peu étudié en TAL et pour lequel il existe peu de ressources annotées : l'astrophysique. Pour cela, nous proposons de comparer trois approches d'augmentation de données en utilisant les deux seuls corpus du domaine disponibles, tous deux en anglais : DEAL (Grezes *et al.*, 2022) et TDAC (Alkan *et al.*, 2022).

Notre étude a mis en evidénce la présence d'un effet de seuil, au-delà duquel l'ajout de données artificielles peut dégrader les performances des systèmes de détection d'entités nommées. L'évaluation des approches d'augmentation sur les deux corpus du domaine astrophysique montrent que les performances varient en fonction de la méthode employée ainsi que du modèle de langue utilisé pour générer les données artificielles. Nous décrivons notre méthode dans la Section 3. Nous présentons nos résultats expérimentaux dans la Section 4, et les analysons en Section 5.1.

2 Recherches connexes

2.1 Méthodes existantes

Explorée principalement en vision par ordinateur (Shorten & Khoshgoftaar, 2019), l'augmentation de données consiste à générer de nouvelles données artificielles pour l'entraînement des modèles, sans avoir à en recueillir davantage de façon manuelle. Ces dernières années, cette technique fait l'objet d'une attention croissante par les chercheurs en TAL, notamment pour la traduction (Wang *et al.*, 2018), les systèmes de questions-réponses (Yang *et al.*, 2019), la classification (Claveau *et al.*, 2021) et le résumé automatique de texte (Pasunuru *et al.*, 2021).

Les méthodes d'augmentation peuvent se diviser en plusieurs familles (Feng *et al.*, 2021). Une première famille regroupant les méthodes à base de règles, utilisant des ressources linguistiques comme WordNet (Miller, 1994) pour remplacer des mots d'une phrase, ou encore la permutation de mots (Wei & Zou, 2019). La deuxième famille repose sur les modèles de langue, utilisant des techniques de rétrotraduction par exemple (Yu *et al.*, 2018). Le remplacement des mots d'une phrase peut également s'effectuer en utilisant un modèle de langue masqué (MLM). L'entraînement d'un modèle de langue masqué constitue la phase de pré-entraînement des modèles de langue contextualisés de type BERT (Devlin *et al.*, 2019). Cette tâche consiste à masquer aléatoirement un mot dans le texte d'entrée avec le token `[MASK]`, puis à prédire le mot masqué. L'insertion du mot prédit dans le texte permet alors de constituer une séquence artificielle à partir de la séquence d'origine. Une troisième famille d'augmentation existe, reposant cette fois sur des techniques d'interpolation (Zhang *et al.*, 2018, 2020). Ces méthodes consistent à générer des données artificielles en interpolant entre des échantillons de données existants. Cela peut se faire en prenant par exemple deux séquences d'entrée (sélectionnées selon leur similarité ou leur pertinence), puis en générant un échantillon artificiel qui se situe entre eux.

2.2 Attention moindre pour la reconnaissance d'entités nommées

Si elle améliore les performances de certaines tâches de TAL telles que le traduction automatique (Nguyen *et al.*, 2020) et les systèmes de questions-réponses (Yu *et al.*, 2018), l'augmentation de données est plus difficile à mettre en œuvre pour la détection d'entités. De récentes études (Dai & Adel, 2020) soulignent que ces méthodes peuvent générer des données artificielles erronées lorsqu'elles sont appliquées sur des tâches au niveau du token. En effet, l'utilisation d'un MLM pour remplacer une entité peut conduire à des erreurs lors de l'affinage de systèmes de reconnaissance d'entités nommées si le type du mot prédit ne correspond pas au type du mot masqué. En raison de cette difficulté, l'augmentation de données pour la REN a fait l'objet d'une attention moindre comparée à d'autres tâches. Pour pallier ce problème, Zhou *et al.* (2022) ont proposé la méthode MELM : *Masked Entity Language Modeling*, qui vise à insérer autour de chaque token masqué le type qui lui correspond lors de la phase de pré-entraînement (par exemple, le token spécial `[B-Ins]` ou `[I-Ins]` autour des tokens d'une entité de la classe `Instrument`). Par conséquent, lors de l'apprentissage, la prédiction du token masqué est conditionnée à la fois par son contexte et par son type, réduisant le risque de non-correspondance entre le token prédit et la classe.

L'astrophysique est un domaine de spécialité générant une quantité significative de documents à analyser, mais possédant en revanche très peu de corpus annotés en entités nommées. L'un de ces corpus, publié très récemment, a servi dans la campagne d'évaluation internationale DEAL sur la reconnaissance d'entités nommées dans des articles d'astrophysique (Grezes *et al.*, 2022). Parmi les travaux qui se sont intéressés à la détection d'entités nommées en astrophysique (Becker *et al.*, 2005; Hachey *et al.*, 2005; Murphy *et al.*, 2006), une seule méthode d'augmentation de données a été proposée par l'un des participants de cette campagne d'évaluation (Huang, 2022). L'auteur utilise des modèles pré-entraînés spécifiques (He *et al.*, 2021; Berquand *et al.*, 2021) à base d'adapteurs (Houlsby *et al.*, 2019) pour l'augmentation de données, lui permettant d'atteindre une F-mesure de 0,7799 sur le jeu de test du corpus DEAL.

3 Protocole expérimental pour l'augmentation de données

3.1 Présentation générale des corpus

Nous utilisons les deux seuls corpus annotés et disponibles du domaine astrophysique : le corpus DEAL (Grezes *et al.*, 2022) et le corpus TDAC (Alkan *et al.*, 2022). Ces ensembles de données proviennent de sources différentes et sont en anglais, car la communauté astrophysique, composée d'amateurs et de professionnels utilise principalement l'anglais pour communiquer.

Le corpus DEAL Ce corpus est constitué de fragments d'articles scientifiques en astrophysique générale. Il a été annoté en entités nommées pour la campagne d'évaluation DEAL (*Detecting Entities in the Astrophysics Literature*) et se compose de trois ensembles : entraînement, développement et test, comprenant respectivement 1753, 1366 et 2505 documents. Le corpus est accessible sur HuggingFace [1].

1. https://huggingface.co/datasets/adsabs/WIESP2022-NER/tree/main

Le corpus TDAC Ce corpus se compose de rapports d'observation (courts messages textuels) qui constituent l'une des sources premières de partage d'informations entre astronomes. A la différence du corpus DEAL, ce corpus se focalise uniquement sur les phénomènes cosmiques dits « transitoires [2] » (Neronov, 2019), possédant ainsi un vocabulaire et un discours spécifiques que nous ne retrouvons pas nécessairement dans le corpus DEAL. Le corpus TDAC est accessible sur GitHub [3]. L'une des limites de ce corpus concerne son nombre limité de documents annotés disponibles : 75 rapports d'observations, dont 59 pour l'entraînement et 16 pour le test. Le tableau 1 fournit quelques statistiques concernant ces deux corpus.

Corpus	Nb classes	Nb tokens	tokens annotés	Long. moyenne (tokens)
DEAL	31	1 815 237	337 663	322
TDAC	28	26 133	4526	256

TABLE 1 – Statistiques des deux corpus d'étude.

Entités nommées et particularités des textes en astrophysique Bien que ces corpus soient de sources différentes, les catégories définies ainsi que les schémas d'annotation sont identiques. Le guide d'annotation comprend 31 entités nommées au total et couvre les entités d'intérêt du domaine : installations astronomiques, objets célestes, coordonnées, formules ou encore techniques d'observation. Une liste détaillée des classes est disponible sur le dépôt HuggingFace [4]. La figure 1 montre la proportion des entités nommées annotées dans chacun des deux corpus.

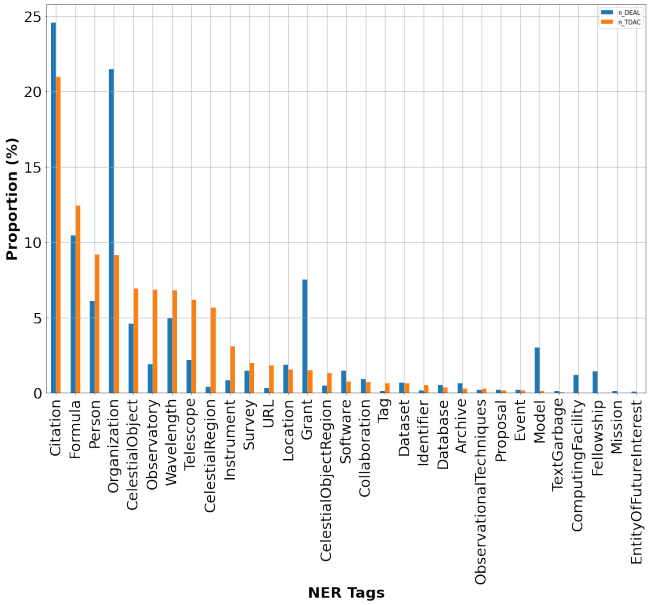


FIGURE 1 – Proportions des entités nommées dans le corpus DEAL (en bleu) et dans TDAC (en orange).

2. Les phénomènes transitoires sont de violentes explosions de courtes durées telles que les explosions de supernovae, les sursauts gamma, ou encore les jets de particules en provenance de certaines galaxies.

3. https://github.com/AtillaKaanAlkan/TDAC

4. https://huggingface.co/datasets/adsabs/WIESP2022-NER/blob/main/tag_definitions.md

La répartition des classes au sein des deux corpus n'est pas similaire. En effet, dans le corpus TDAC, les catégories les plus fréquentes sont par exemple `Formula`, `CelestialObject`, `Observatory` ou `CelestialRegion`. Il s'agit de catégories spécifiques au domaine astrophysique. La plupart de ces classes spécifiques sont moins présentes dans le corpus DEAL, dans lequel on retrouve principalement des classes du type : `Citation`, `Organization`, `Grant` ou `Person`, qui semblent être des catégories d'entités nommées plus génériques dans les articles scientifiques.

3.2 Description des méthodes d'augmentation

Nous masquons aléatoirement 70 % des entités nommées d'une séquence sans changer les tokens de type « O ». La valeur de 70 % a été suggérée par Zhou *et al.* (2022) pour la méthode MELM à l'issue d'une recherche par grille (*grid search*). Nous avons en plus fait le choix que si un token faisant partie d'une portion annotée est masqué, alors l'intégralité de cette portion est remplacée. La figure 2 schématise nos trois méthodes.

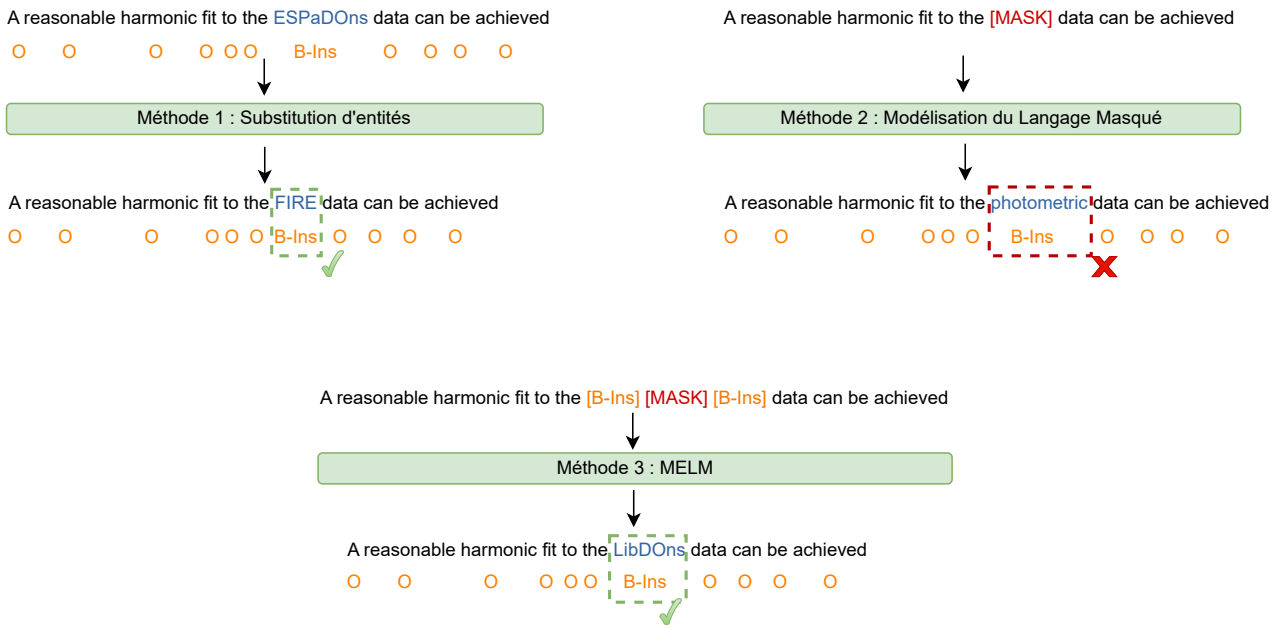


FIGURE 2 – Illustration des trois méthodes d'augmentation proposées.

Méthode 1 : Substitution d'entités à base de règles La première méthode que nous explorons consiste à remplacer aléatoirement une entité d'une classe donnée par une autre entité de cette même classe du corpus d'entraînement. Par exemple, nous remplaçons un nom d'instrument de mesure comme « ESPaDOnS » de la classe `Instrument` par « NIRCam » de cette même classe, créant ainsi une donnée augmentée. Il s'agit donc d'une méthode simple d'augmentation à base de règles.

Méthode 2 : Modèle de langue masqué (MLM) La deuxième méthode que nous proposons consiste à utiliser des modèles de langue masqué. Afin d'évaluer l'impact de l'augmentation de données, nous utilisons trois modèles : astroBERT (Grezes *et al.*, 2021), SciBERT (Beltagy *et al.*, 2019) et RoBERTa (Zhuang *et al.*, 2021). Ces modèles se distinguent par leur corpus de pré-entraînement.

Alors qu'astroBERT est conçu pour l'analyse de textes en astrophysique, SciBERT repose sur des textes scientifiques, tandis que RoBERTa est un modèle pré-entraîné sur des corpus de langue générale. Nous masquons aléatoirement des entités du corpus et utilisons la prédiction du mot masqué pour produire nos données artificielles.

Méthode 3 : Modèle de langue d'entités masquées (MELM) Comme évoqué dans les travaux connexes, l'usage d'un modèle de langue masqué peut générer des erreurs de type. Dans cette troisième méthode, nous proposons d'appliquer la méthode MELM pour la prédiction de nouveaux mots, et constituer ainsi nos données artificielles. Comme illustré sur la figure 2, nous ajoutons autour des tokens masqués les étiquettes correspondantes. Le modèle astroBERT étant spécifique au domaine de spécialité et la méthode MELM s'appuyant sur les classes d'entités relatives au domaine astrophysique, nous avons étudié l'impact de cette méthode au moyen d'astroBERT principalement. Nous pré-entraînons astroBERT sur la tâche de prédiction de mot masqué avec les tokens spéciaux ajoutés en utilisant les échantillons d'entraînement respectifs des deux corpus : 1753 documents pour DEAL, et 59 rapports d'observation pour TDAC pendant 20 époques, comme conseillé par Zhou *et al.* (2022).

4 Expériences pour la reconnaissance d'entités nommées

4.1 Configurations

Modèles et hyperparamètres Nous avons affiné astroBERT (Grezes *et al.*, 2021) sur une tâche de REN sur 15 époques, avec un taux d'apprentissage $\alpha = 2 \times 10^{-5}$ et une taille de lot d'entraînement (*training batch size)* de 4. Les expériences ont été réitérées cinq fois avec des valeurs d'amorces différentes ($seeds = [0, 123, 762, 5000, 6822]$) choisies aléatoirement.

Variation du taux d'augmentation Amalvy *et al.* (2022) ont montré qu'au delà d'un certain seuil, l'augmentation de données ne présentait plus d'intérêt et engendrait une baisse des performances, principalement due à un sur-apprentissage. C'est pourquoi nous avons analysé l'impact de l'augmentation de données sur les performances des systèmes de détection d'entités nommées en faisant varier la quantité d'exemples artificiels ajoutés au corpus d'entraînement d'origine. Plus précisément, nous avons examiné des augmentations par incrément de 25 %.

4.2 Résultats

Afin d'évaluer nos systèmes de détection d'entités nommées, nous calculons la précision (P), le rappel (R) et la F-mesure (F) en suivant la méthode CoNLL-2003 (Tjong Kim Sang & De Meulder, 2003). Nous présentons les résultats obtenus sur les jeux de test des corpus DEAL et TDAC par un modèle de détection d'entités astroBERT entraîné sur des données étendues avec chacune des trois méthodes d'augmentation de données que nous avons présentées : substitution d'entités (tableau 2), modèle de langue masqué (tableau 3) et MELM (tableau 4). Pour évaluer l'intérêt de l'augmentation de données, nous rappelons sur la première ligne de chaque tableau les performances du système entraîné uniquement sur les données d'origine (sans augmentation).

Taux	DEAL			TDAC		
	P	R	F	P	R	F
∅	0,799	**0,834**	**0,816** (0,002)	0,666	0,737	0,700 (0,003)
25 %	**0,805**	0,827	0,816 (0,003)	0,641	0,726	0,681 (0,018)
50 %	0,800	0,823	0,811 (0,003)	0,663	0,728	0,694 (0,014)
75 %	0,801	0,825	0,813 (0,001)	0,667	0,734	0,699 (0,009)
100 %	0,789	0,826	0,807 (0,003)	**0,677**	**0,749**	**0,711 (0,004)**

TABLE 2 – Impact de l'augmentation de données par la méthode de substitution d'entités, évaluée sur une tâche de repérage d'entités nommées, avec comparaison sans augmentation de données (∅). Pour chaque configuration, nous affichons la moyenne (et l'écart type) des 5 affinages réalisés. Pour des raisons de visibilité, nous indiquons l'écart type uniquement pour la F-mesure.

Modèle et taux d'augmentation		DEAL			TDAC		
		P	R	F	P	R	F
	∅	0,799	0,834	0,816 (0,002)	0,666	0,737	0,700 (0,003)
astroBERT	25 %	0,799	0,829	0,814 (0,005)	0,647	0,728	0,685 (0,007)
	50 %	0,795	0,824	0,809 (0,005)	0,664	0,729	0,695 (0,009)
	75 %	0,789	0,824	0,806 (0,004)	0,670	0,748	0,706 (0,016)
	100 %	0,788	0,820	0,804 (0,002)	0,680	0,749	0,713 (0,009)
SciBERT	25 %	0,796	0,831	0,813 (0,004)	0,631	0,718	0,671 (0,014)
	50 %	0,794	0,828	0,810 (0,002)	**0,693**	**0,749**	**0,720 (0,016)**
	75 %	0,788	0,821	0,804 (0,006)	0,677	0,744	0,709 (0,013)
	100 %	0,776	0,822	0,798 (0.003)	0,688	0,745	0,715 (0,008)
RoBERTa	25 %	0,799	0,836	0,817 (0,001)	0,624	0,710	0,664 (0,011)
	50 %	0,794	**0,839**	0,816 (0,003)	0,635	0,711	0,671 (0,014)
	75 %	**0,806**	0,835	**0,820 (0,003)**	0,674	0,728	0,700 (0,014)
	100 %	0,798	0,831	0,814 (0,003)	0,639	0,707	0,671 (0,008)

TABLE 3 – Impact de l'augmentation de données par la méthode MLM, en fonction du modèle de langue masqué utilisé (astroBERT, SciBERT, RoBERTa) pour générer les données artificielles et du taux d'augmentation (de 25,0 % à 100 % par incrément de 25,0 %), évaluée sur une tâche de repérage d'entités nommées, avec comparaison sans augmentation de données (∅). Pour chaque configuration, nous affichons la moyenne (et l'écart type) des 5 affinages réalisés. Pour des raisons de visibilité, nous indiquons l'écart type uniquement pour la F-mesure.

5 Analyse et discussion

5.1 Impact de la méthode d'augmentation et du modèle de langue

Les trois méthodes d'augmentation proposées fournissent des résultats assez proches. Néanmoins, dans le cadre de nos expériences, nous constatons que les approches basées sur l'utilisation de modèles de langue (méthodes 2 et 3) permettent d'obtenir de meilleurs scores qu'une approche de substitution à base de règles. Nous gagnons jusqu'à deux points de F-mesure avec les méthodes 2 et 3 (tableaux 3 et 4) sur le corpus TDAC, et 1 point sur le corpus DEAL, tandis que la méthode à base de règles (tableau 2) nous permet de gagner 1 point de F-mesure sur le corpus TDAC, et aucune amélioration

Modèle et taux d'augmentation		DEAL			TDAC		
		P	**R**	**F**	**P**	**R**	**F**
	$\emptyset$	0,799	0,834	0,816 (0, 002)	0,666	0,737	0,700 (0, 003)
astroBERT	25 %	0,798	0,835	0,816 (0, 005)	0,664	0,732	0,696 (0, 008)
	50 %	0,796	0,834	0,815 (0, 003)	0,667	0,738	0,701 (0, 004)
	75 %	0,797	0,835	0,816 (0, 004)	0,668	0,738	0,701 (0, 004)
	100 %	0,797	0,836	0,816 (0, 004)	0,669	0,738	0,702 (0, 005)

TABLE 4 – Impact de l'augmentation de données par la méthode MELM, en fonction du taux d'augmentation (de 25,0 % à 100 % par incrément de 25,0 %),évaluée sur une tâche de repérage d'entités nommées, avec comparaison sans augmentation de données ($\emptyset$). Pour chaque configuration, nous affichons la moyenne (et l'écart type) des 5 affinages réalisés. Pour des raisons de visibilité, nous indiquons l'écart type uniquement pour la F-mesure.

n'est constatée sur le corpus DEAL.

Cette observation peut s'expliquer par la diversité générée grâce aux modèles de langue. En effet, ces derniers permettent d'offrir une plus grande diversité dans la génération de nouvelles entités, contrairement à la méthode 1 qui, reste limitée aux entités présentes dans le corpus d'entraînement lors de la génération de données.

Toujours en lien avec la diversité, nous constatons dans le tableau 3 que l'utilisation d'un modèle de langue générale (RoBERTa) sur DEAL et d'un modèle entraîné sur des textes scientifiques (SciBERT) sur TDAC pour générer des données artificielles, permettent d'obtenir de meilleurs résultats que l'utilisation d'astroBERT. Il semble donc y avoir un impact du corpus d'entrainement du modèle de langue sur l'augmentation de données : un modèle de langue pré-entraîné sur un corpus plus général, permet de générer une plus grande diversité. Nous envisageons donc de poursuivre la comparaison de la méthode MELM avec les deux autres modèles RoBERTa et SciBERT.

De plus, nous estimons que la capacité de prise en compte du contexte des modèles de langue lors de la génération de données peut également être une des raisons conduisant à de meilleurs résultats.

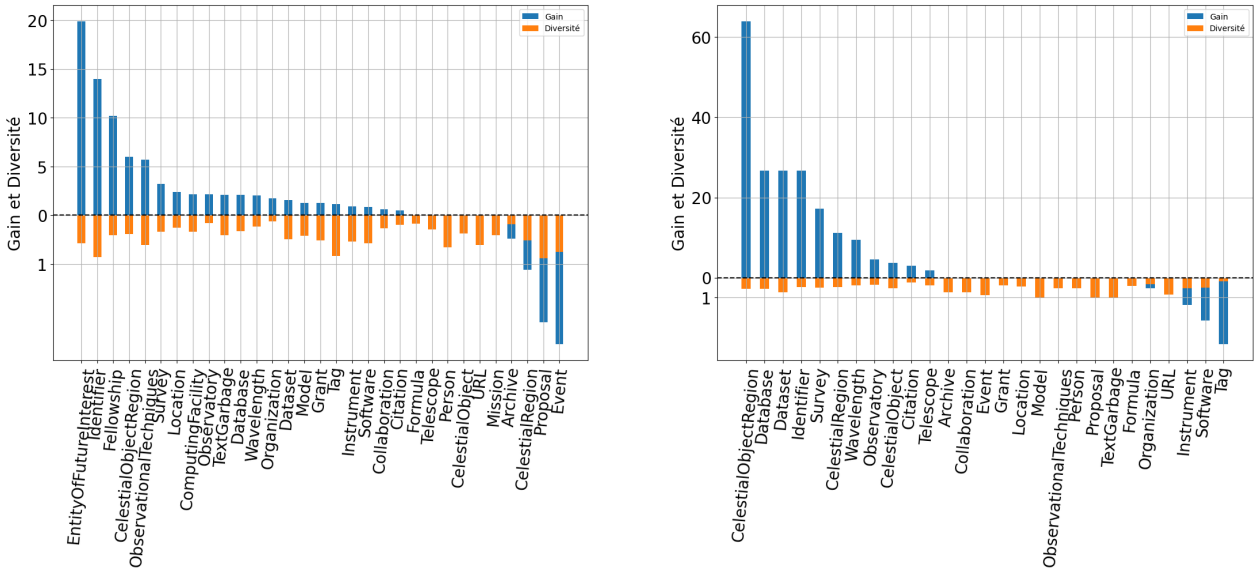
5.2 Effet de seuil

L'augmentation par incrément nous permet d'observer un effet de seuil, qui est également constaté dans les travaux de Amalvy *et al.* (2022). En effet, peu importe la méthode d'augmentation proposée, l'ajout de données artificielles au delà d'une certaine quantité ne présente plus d'intérêt et dégrade la F-mesure. Dans nos expériences, ce seuil varie selon la méthode employée et en fonction du modèle de langue utilisé. Il semblerait que plus le modèle de langue est général, plus le seuil est élevé. Ceci peut s'expliquer par le fait que les données générées avec astroBERT sont très proches des données d'origines et moins diversifiées que celles produites par RoBERTa, conduisant à un sur-apprentissage.

Par ailleurs, au vu des résultats obtenus, l'augmentation de données semble plus efficace lorsque appliquée sur de petits corpus, ce qui rejoint également la conclusion de Dai & Adel (2020). En effet, il est possible de gagner jusqu'à plus de 2 points de F-mesure sur le corpus TDAC (tableau 3), alors que sur le corpus DEAL, plus grand, nous gagnons seulement 1 point (tableaux 3 et 4) toutes méthodes confondues.

5.3 Répartition des gains et diversité des classes

En considérant les méthodes ayant donné les meilleurs scores globaux en terme de F-mesure, nous avons analysé pour chacune des classes (types d'entités) s'il existait un lien entre le gain obtenu (diminution ou augmentation de la F-mesure) et la diversité de la classe (nombre de formes de surface différentes). Le calcul du coefficient de corrélation de Pearson [5] (c_p) montre qu'il y a une faible corrélation négative entre le gain et la diversité. En effet, les coefficients sont relativement faibles : $c_p = -0{,}062$ pour le corpus DEAL, et $c_p = 0{,}023$ pour le corpus TDAC. La figure 3 montre que le gain est réparti sur différentes classes.



(a) Corpus DEAL

(b) Corpus TDAC

FIGURE 3 – Répartition des gains (en bleu) et diversité (en orange) sur les deux corpus.

Dans l'ensemble, la majeure partie des classes bénéficient d'une augmentation de la F-mesure avec une amélioration significative sur les classes du domaine astrophysique. En effet, l'ajout de 25 % de données artificielles au corpus DEAL avec la méthode MELM permet au système de REN de mieux repérer certains concepts dans le texte tels que les coordonnées d'objets célestes et les techniques utilisées lors des observations : 6 points pour la classe `CelestialObjectRegion`, et 5,7 pour la classe `ObservationalTechniques`. L'augmentation de données présente également un intérêt pour le repérage d'installations astronomiques (`Observatory` :2,13) et des longeurs d'ondes (`Wavelength` : 2,02). Nous constatons également des améliorations dans la détection des coordonnées sur le corpus TDAC (`CelestialObjectRegion` : 63,9) et des noms d'objets astrophysiques (`CelestialObject` :3,71).

6 Conclusion et perspectives

Dans cet article, nous avons étudié l'intérêt de l'augmentation de données pour entraîner des systèmes de repérage d'entités nommées. Les méthodes proposées ont été appliquées à un domaine de spécialité

5. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html

peu étudié en TAL : l'astrophysique, en utilisant deux corpus existants. Nous avons exploré l'efficacité de trois méthodes d'augmentation permettant de limiter les erreurs token-étiquette lors de la génération de données.

Notre étude a permis d'identifier les classes qui bénéficient le plus de l'augmentation et de mettre en évidence que les performances varient selon la méthode d'augmentation employée, en fonction du modèle de langue utilisé.

Ces résultats ouvrent la voie à plusieurs perspectives de recherche. Tout d'abord, il pourrait être bénéfique de cibler spécifiquement les classes à augmenter plutôt que de procéder à une augmentation aléatoire. En effet, cibler les classes à augmenter sur le corpus DEAL pourrait éviter une diminution des performances sur deux classes importantes du domaine astrophysique : `CelestialRegion` ($-5{,}59$), et `CelestialObject` ($-0{,}29$).

En outre, il serait intéressant de trouver une approche permettant de diversifier la structure des documents et de modifier le contexte local autour des entités. En effet, bien que les méthodes comparées dans cet article permettent d'augmenter le nombre d'entités du corpus et leur diversité (selon le modèle utilisé), elles ne changent pas la structure des documents et ne modifient pas le contexte, ce qui génère des données artificielles proches des données d'origine, pouvant conduire à un risque de sur-apprentissage qui expliquerait pourquoi les performances des systèmes peuvent se dégrader au-delà d'un certain seuil.

Une solution serait de remplacer les tokens de type « O » (les adjectifs et les adverbes par exemple) autour des entités annotées afin de pouvoir modifier le contexte localement. Enfin, il pourrait être utile de tester les méthodes à base d'interpolation (Zhang *et al.*, 2020) et de combiner plusieurs approches d'augmentation pour améliorer davantage les performances.

Remerciements

Nous remercions chaleureusement Fabian Schüssler (IRFU, CEA, Université Paris-Saclay), astrophysicien, pour son accompagnement, son expertise et ses nombreux conseils dans le cadre de ces travaux de recherches.

Références

ALKAN A. K., GROUIN C., SCHUSSLER F. & ZWEIGENBAUM P. (2022). TDAC, the first corpus in time-domain astrophysics : Analysis and first experiments on named entity recognition. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, p. 131–139, Online : Association for Computational Linguistics.

AMALVY A., LABATUT V. & DUFOUR R. (2022). Remplacement de mentions pour l'adaptation d'un corpus de reconnaissance d'entités nommées à un domaine cible (Mention replacement for adapting a named entity recognition dataset to a target domain). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 198–205, Avignon, France : ATALA.

BANERJEE P. S., CHAKRABORTY B., TRIPATHI D., GUPTA H. & KUMAR S. S. (2019). A information retrieval based on question and answering and ner for unstructured information without using sql. *Wirel. Pers. Commun.*, **108**(3), 1909–1931. DOI : 10.1007/s11277-019-06501-z.

BECKER M., HACHEY B., ALEX B. & GROVER C. (2005). Optimising selective sampling for bootstrapping named entity recognition. In *In Proceedings of the ICML Workshop on Learning with Multiple Views*, p. 5–11.

BELTAGY I., LO K. & COHAN A. (2019). SciBERT : A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3615–3620, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1371.

BERQUAND A., DARM P. & RICCARDI A. (2021). Spacetransformers : Language modeling for space systems. *IEEE Access*, **9**, 133111–133122. DOI : 10.1109/ACCESS.2021.3115659.

CLAVEAU V., CHAFFIN A. & KIJAK E. (2021). La génération de textes artificiels en substitution ou en complément de données d'apprentissage (Generating artificial texts as substitution or complement of training data). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 37–49, Lille, France : ATALA.

DAI X. & ADEL H. (2020). An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 3861–3867, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.343.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

FENG S. Y., GANGAL V., WEI J., CHANDAR S., VOSOUGHI S., MITAMURA T. & HOVY E. H. (2021). A survey of data augmentation approaches for NLP. *CoRR*, **abs/2105.03075**.

GREZES F., BLANCO-CUARESMA S., ACCOMAZZI A., KURTZ M. J., SHAPURIAN G., HENNEKEN E., GRANT C. S., THOMPSON D. M., CHYLA R., MCDONALD S., HOSTETLER T. W., TEMPLETON M. R., LOCKHART K. E., MARTINOVIC N., CHEN S., TANNER C. & PROTOPAPAS P. (2021). Building astroBERT, a language model for astronomy & astrophysics. DOI : 10.48550/ARXIV.2112.00590.

GREZES F., BLANCO-CUARESMA S., ALLEN T. & GHOSAL T. (2022). Overview of the first shared task on detecting entities in the astrophysics literature (DEAL). In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, p. 1–7, Online : Association for Computational Linguistics.

GRISHMAN R. & SUNDHEIM B. (1996). Message Understanding Conference- 6 : A brief history. In *COLING 1996 Volume 1 : The 16th International Conference on Computational Linguistics*.

HACHEY B., ALEX B. & BECKER M. (2005). Investigating the effects of selective sampling on the annotation task. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, p. 144–151, Ann Arbor, Michigan : Association for Computational Linguistics.

HE P., GAO J. & CHEN W. (2021). Debertav3 : Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, **abs/2111.09543**.

HOULSBY N., GIURGIU A., JASTRZEBSKI S., MORRONE B., DE LAROUSSILHE Q., GESMUNDO A., ATTARIYAN M. & GELLY S. (2019). Parameter-efficient transfer learning for NLP. In K. CHAUDHURI & R. SALAKHUTDINOV, Éds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 de *Proceedings of Machine Learning Research*, p. 2790–2799 : PMLR.

HUANG P.-W. (2022). Domain specific augmentations as low cost teachers for large students. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, p. 84–90, Online : Association for Computational Linguistics.

MILLER G. A. (1994). WordNet : A lexical database for English. In *Human Language Technology : Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

MOLLÁ ALIOD D., VAN ZAANEN M. & SMITH D. (2006). Named entity recognition for question answering. In L. CAVEDON & I. ZUKERMAN, Éds., *Proceedings of the Australasian Language Technology Workshop, ALTA 2006, Sydney, Australia, November 30-December 1, 2006*, p. 51–58 : Australasian Language Technology Association.

MURPHY T., MCINTOSH T. & CURRAN J. R. (2006). Named entity recognition for astronomy literature. In *Proceedings of the Australasian Language Technology Workshop 2006*, p. 59–66, Sydney, Australia.

NERONOV A. (2019). Introduction to multi-messenger astronomy. *Journal of Physics : Conference Series*, **1263**(1), 012001. DOI : 10.1088/1742-6596/1263/1/012001.

NGUYEN X., JOTY S. R., WU K. & AW A. T. (2020). Data diversification : A simple strategy for neural machine translation. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

PASUNURU R., CELIKYILMAZ A., GALLEY M., XIONG C., ZHANG Y., BANSAL M. & GAO J. (2021). Data augmentation for abstractive query-focused multi-document summarization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, p. 13666–13674 : AAAI Press.

SHORTEN C. & KHOSHGOFTAAR T. M. (2019). A survey on image data augmentation for deep learning. *J. Big Data*, **6**, 60. DOI : 10.1186/s40537-019-0197-0.

TJONG KIM SANG E. F. & DE MEULDER F. (2003). Introduction to the CoNLL-2003 shared task : Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, p. 142–147.

WANG X., PHAM H., DAI Z. & NEUBIG G. (2018). SwitchOut : an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 856–861, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1100.

WEI J. & ZOU K. (2019). EDA : Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 6382–6388, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1670.

YANG W., XIE Y., TAN L., XIONG K., LI M. & LIN J. (2019). Data augmentation for BERT fine-tuning in open-domain question answering. *CoRR*, **abs/1904.06652**.

YU A. W., DOHAN D., LUONG M., ZHAO R., CHEN K., NOROUZI M. & LE Q. V. (2018). QANet : Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* : OpenReview.net.

ZHANG H., CISSÉ M., DAUPHIN Y. N. & LOPEZ-PAZ D. (2018). mixup : Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* : OpenReview.net.

ZHANG R., YU Y. & ZHANG C. (2020). SeqMix : Augmenting active sequence labeling via sequence mixup. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 8566–8579, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.691.

ZHOU R., LI X., HE R., BING L., CAMBRIA E., SI L. & MIAO C. (2022). MELM : Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2251–2262, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.160.

ZHUANG L., WAYNE L., YA S. & JUN Z. (2021). A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, p. 1218–1227, Huhhot, China : Chinese Information Processing Society of China.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Towards a Robust Detection of Language Model-Generated Text: Is ChatGPT that Easy to Detect?

Wissam Antoun Virginie Mouilleron Benoît Sagot Djamé Seddah
Inria, Paris, France
firstname.lastname@inria.fr

ABSTRACT

Recent advances in natural language processing (NLP) have led to the development of large language models (LLMs) such as ChatGPT. This paper proposes a methodology for developing and evaluating ChatGPT detectors for French text, with a focus on investigating their robustness on out-of-domain data and against common attack schemes. The proposed method involves translating an English dataset into French and training a classifier on the translated data. Results show that the detectors can effectively detect ChatGPT-generated text, with a degree of robustness against basic attack techniques in in-domain settings. However, vulnerabilities are evident in out-of-domain contexts, highlighting the challenge of detecting adversarial text. The study emphasizes caution when applying in-domain testing results to a wider variety of content. We provide our translated datasets and models as open-source resources.[1]

RÉSUMÉ

Vers une détection robuste de texte généré par un modèle de langue : ChatGPT est-il si facile à détecter ?

Les récents progrès en traitement automatique des langues (TAL) ont conduit au développement de grands modèles de langage (*Large Language Models*, LLM) tels que ChatGPT. Cet article propose une méthodologie pour développer et évaluer des modèles de détection de contenus en français produits par ChatGPT, en mettant l'accent sur leur robustesse face à des données hors domaine et face à des attaques classiques. La méthode proposée consiste à traduire un ensemble de données anglaises en français et à entraîner un classificateur sur les données traduites. Les résultats montrent que les détecteurs peuvent efficacement détecter le texte généré par ChatGPT, avec un bon niveau de robustesse contre les techniques usuelles d'attaque sans changement de domaine. Cependant, les résultats sont moins bons dans les contextes hors domaine, soulignant le défi que constitue toujours de contenus adversariaux. Notre étude souligne l'importance de rester prudent lorsque l'on cherche à généraliser des résultats obtenus sans changement de domaine à une plus grande variété de contenus. Tous nos jeux de données et nos modèles sont distribués librement.

KEYWORDS: ChatGPT, text generation, detection of machine-generated text, robustness.

MOTS-CLÉS : ChatGPT, génération de texte, détection de texte généré par machine, robustesse.

[1] https://gitlab.inria.fr/wantoun/robust-chatgpt-detection

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 14–27.

1 Introduction

Advances in natural language processing (NLP) have been driven mainly by scaling up the size of pre-trained language models, along with the amount of data and compute required for training (Raffel *et al.*, 2020; Radford *et al.*, 2019; Rae *et al.*, 2021; Fedus *et al.*, 2021; Hoffmann *et al.*, 2022). OpenAI recently released ChatGPT, a text generation model with conversational capabilities. The model is based on GPT3.5 which is a version of GPT3 (Brown *et al.*, 2020) first fine-tuned on code then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) (Christiano *et al.*, 2017; Stiennon *et al.*, 2020), a method previously demonstrated by OpenAI with InstructGPT (Ouyang *et al.*, 2022). This fine-tuning process contributes not only to the model's knowledge but also simplifies the model's interface compared to GPT3, which necessitated substantial *prompt engineering* to achieve satisfactory outcomes, and hence facilitating the extraction and application of that built-in knowledge.

As a result of these significant performance improvements, ChatGPT and other large language models have gained much popularity in the media and in the social context, often without fully understanding the underlying limitations of the models – e.g., the possibility of generating hateful, hateful, toxic, or disrespectful content (Bender *et al.*, 2021; McGuffie & Newhouse, 2020; Weidinger *et al.*, 2021). Another potential misuse of LLMs or ChatGPT is industrializing radicalization and harmful propaganda which poses a significant and unconventional threat to civil society.

In response to the mounting concerns surrounding potential misuse, numerous researchers are now exploring various strategies to mitigate associated risks. For example, some have proposed watermarking techniques to trace the origin of generated text[2], while others are developing methods to detect and flag text generated by these models. Of particular interest, a recent study by Guo *et al.* (2023) investigated the text generation capabilities of ChatGPT and its proximity to human-generated text. To create a dataset of ChatGPT-generated text, the authors leveraged pre-existing question-answering datasets in both English and Chinese, using the questions as prompts to generate responses from the model. In addition, the authors conducted a linguistic analysis to compare the output generated by ChatGPT with human-written text, and they also developed a detector to distinguish between ChatGPT-generated text and human-written text by fine-tuning a separate language model on a dataset containing both types of text.

The aim of this research is to explore the development of ChatGPT detectors in multiple languages, along with evaluating their robustness on out-of-domain text, we selected French as the language of interest. Therefore, we propose a methodology that involves translating the English dataset into French and subsequently training a classifier on the translated data. We conducted a series of evaluations in a monolingual and multilingual setting on both in-domain and out-of-domain data. The in-domain data consisted of text generated by ChatGPT using prompts related to the topics covered in the training dataset. The out-of-domain data included text generated in French by ChatGPT and Bing, a search engine powered by ChatGPT[3], which has access to a broader range of internet content and may generate text on a wider range of topics than ChatGPT. Given that Wolff & Wolff (2020) demonstrated the vulnerability of BERT-based detectors for GPT-2 against basic attack schemes, such as substituting characters with homoglyphs or misspelled words, we also evaluated the robustness of our models against these types of attacks. Furthermore, we hypothesize that the detector models we trained rely

[2]At the time of writing, OpenAI was working on a tool to statistically watermark text generated by GPT-like models according to Scott Aaronson, a guest researcher at OpenAI https://scottaaronson.blog/?p=6823

[3]https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/

heavily on the didactic response style of ChatGPT to distinguish between human-generated content.

The contributions of this study can be summarized as follows:

- We build upon the work of Guo *et al.* (2023) and propose a methodology to develop ChatGPT detectors in multiple languages, focusing on French as a case study.
- We evaluated the performance of the ChatGPT detectors in both a monolingual and multilingual setting. Specifically, we trained and evaluated our models on both the English and French datasets, as well as on a combined dataset containing both languages.
- We investigate the generalizability of our detector by testing its performance on out-of-domain data.
- We evaluate the robustness of our models against common attack schemes, such as substituting characters with homoglyphs or misspelled words. This is an important aspect to consider in the deployment of ChatGPT detectors, as attackers may attempt to evade detection by modifying the generated text in subtle ways.
- We investigated the reliance of detector models on ChatGPT's didactic response style for distinguishing between human-generated content.
- We release all translated datasets and models as open source to encourage further research in this area and to enable others to replicate our experiments.

Overall, our work contributes to the growing body of research on developing and evaluating ChatGPT detectors, with a focus on multilinguality, generalizability, and robustness to attacks. Our findings have practical implications for the use of ChatGPT detectors in various settings, including social media platforms, online forums, and chatbots, where the detection of harmful content is critical for maintaining a safe and respectful online environment.

2 Related Works

2.1 Large Language Models

The race to scale up language models to new heights has been a hot topic in recent years. Researchers and tech companies have been competing to develop larger and more powerful models, often breaking records for model size and performance. The trend began with OpenAI's GPT-2 (Radford *et al.*, 2018), which was released in 2019 and featured 1.5 billion parameters. This was quickly followed by Megatron (Shoeybi *et al.*, 2019), a 8.3 billion parameter model, displaying steadily increasing superior zero-shot language model performance, and T5 (Raffel *et al.*, 2020), an 11 billion parameter encoder-decoder model which advanced transfer learning and performance on several closed-book question answering tasks. The release of GPT-3 (Brown *et al.*, 2020), and PaLM (Chowdhery *et al.*, 2022) represented a major milestone in the race to scale up language models, with their unprecedented 175 and 540B billion parameters. Scaling models to such massive scales "unlocks" new emergent capabilities as shown in Chowdhery *et al.* (2022). In November 2022, OpenAI released ChatGPT, a conversational language model based on GPT-3.5 fine-tuned using Reinforcement Learning from Human Feedback (RLHF) (Christiano *et al.*, 2017; Stiennon *et al.*, 2020), a method previously demonstrated by OpenAI with InstructGPT (Ouyang *et al.*, 2022).

2.2 Detecting Synthetic Text

Detecting synthetically generated text is one of the defense mechanisms against harm caused by LLMs. One of the first major explorations of this topic was conducted following the release of GROVER (Zellers *et al.*, 2019) a fake news generator and detector. Since this approach has been shown to work quite well, and as part of their model release strategies (Solaiman *et al.*, 2019), OpenAI also released a GPT2 detector based on a fine-tuned RoBERTa model (Liu *et al.*, 2019), later Fagni *et al.* (2021) demonstrated the performance of another RoBERTa-based detector on machine-generated tweets, Uchendu *et al.* (2020) also used RoBERTa to spot news generated by several language models, while Antoun *et al.* (2021b) created an ELECTRA-based model (Antoun *et al.*, 2021a) to spot articles generated by their AraGPT2. The authors stated that the success of their method was due to the model being pre-trained on the exact same dataset as AraGPT2, and also due to the replaced-token detection pre-training objective (RTD) (Clark *et al.*, 2020), which bears a resemblance to the synthetic text detection objective. Nguyen-Son *et al.* (2021) proposed a detector that uses the text similarity with round-trip translation (TSRT), to detect a machine-translated text from a never before seen translator, and achieved 86.9% detection accuracy. On the other hand, Wolff & Wolff (2020) showed that the RoBERTa GPT-2 detector is vulnerable against simple attack schemes such as substituting characters with homoglyphs or misspelled words. In these cases, the detector's recall went down from 97% to 0.26% and 22.68% respectively. In order to further enhance the accuracy of detecting manipulated news articles that may deceive readers, Jawahar *et al.* (2022) proposed a neural network-based detector that uses factual knowledge via graph convolutional neural network to distinguish between human-written news articles and manipulated news articles that mislead readers.

Following the release of ChatGPT, and with the recognition of the potential risks posed by this highly-capable model, there has been a surge of investigation into methods for detecting ChatGPT-generated text, which led to the commercial release of multiple detector products. However, as the methods used in these products are often not publicly verifiable, this study focuses solely on the academic literature surrounding detection methods. Mitchell *et al.* (2023) proposes a new method called DetectGPT, which leverages negative curvature regions of the model's log probability function and does not require training a separate classifier or watermarking generated text, resulting in a more discriminative approach than existing zero-shot methods for model sample detection. Notebly, Guo *et al.* (2023) created a dataset of ChatGPT responses to queries from diverse sources in English and Chinese. The authors investigated the linguistic and stylistic differences between human and ChatGPT-generated text, in addition to training a variety of classifiers of which a finetuned pretrained language model turned out to be the best.

3 Methodology

3.1 Data Collection

To train and evaluate our ChatGPT detectors, we leveraged the Human ChatGPT Comparison Corpus (HC3) created by Guo *et al.* (2023) which contains both human-written and ChatGPT-generated text in English and Chinese. We primarily focus on the English portion of the dataset as machine translation performs optimally on it. The dataset consists of 24,322 human-written questions and 58546 answers sourced primarily from ELI5 (Fan *et al.*, 2019), WikiQA (Yang *et al.*, 2015), Crawled Wikipedia,

Medical Dialog dataset (Chen *et al.*, 2020), and FiQA (Maia *et al.*, 2018). The authors generated ChatGPT responses using OpenAI's web application,[4] automating the input of questions and scraping the answers with the help of automation testing tools, for a total of 26903 ChatGPT-generated answers.

To create a French dataset, we translated the English dataset using the Google Cloud Translation API. We then split the dataset into three splits train, validation, and test, by first selecting 710 balanced question and answer pairs to be validated, manually annotated[5] and to serve as our test set. We split the rest in an 80/20 split to get the training and validation set.

Furthermore, to assess the ChatGPT detectors' ability to generalize, we manually compiled out-of-domain test data by means of:

- Manually collecting 113 ChatGPT French responses to high-quality translated questions from the test set, referred to as the **ChatGPT-Native** set.
- Using Bing, we manually collect 106 French responses to high-quality translated questions from the test set which we refer to as the **BingGPT**. Given that BinGPT includes source citations in its output, we remove these artifacts from the data (as well as all of its self-referring mentions).
- Randomly sampling 4454 French question-answer pairs from the French subset of the Multilingual FAQ Dataset (MFAQ) (De Bruyn *et al.*, 2021), known as the **FAQ-Rand** set.
- Since the French FAQ data featured in the MFAQ dataset could be machine translated, we create a smaller set from the French FAQ data featured by filtering for .gouv domains, named the **FAQ-Gouv** set.
- 1235 sentences from The French Treebank test set, corpus from Le Monde (Abeillé *et al.*, 2000) articles, which we denote as the **FTB** set.
- Moreover, in order to investigate our hypothesis that the detector relies heavily on the style of ChatGPT and Bing answers to distinguish between human-generated content, we created an additional set of responses to 61 questions. These responses were crafted as "open-book" answers with the same style as those provided by ChatGPT and Bing, resulting in a set of responses that we refer to as the **Adversarial** set.

3.2 Detector Architecture

Our approach fine-tunes pre-trained transformer-based models on our binary classification dataset.

For English, we used two pre-trained transformer models: RoBERTa (Liu *et al.*, 2019) and ELECTRA (Clark *et al.*, 2020). RoBERTa is a variant of BERT (Devlin *et al.*, 2019), trained using masked language modeling. ELECTRA, on the other hand, introduced a new training objective, Replaced Token Detection (RTD), that replaces tokens in the input sequence with tokens generated by another model and then requires the discriminator to distinguish between the replaced and original tokens. We hypothesize that this objective should improve performance since the RTD objective greatly resembles the machine-generated text detection objective.

For French, we used two pre-trained transformer models: CamemBERT (Martin *et al.*, 2020) and CamemBERTa.[6] CamemBERT is a RoBERTa model trained from scratch on French text, while CamemBERTa is based on the DeBERTaV3 (He *et al.*, 2021) architecture and trained from scratch on

[4] https://chat.openai.com/chat

[5] The detailed annotation guideline will be publicly released with our dataset.

[6] The model paper is currently under review and will be released soon.

French text using RTD.

For the multilingual setting, we only fine-tune XLM-R (Conneau *et al.*, 2020), a multilingual RoBERTa model with supports for 100+ languages.[7]

4 Experimental Methodology and Results

4.1 Experiment Design

Motivated by Guo *et al.* (2023), and given that the HC3 dataset comprises question/answer pairs, we investigated three distinct methods for generating dataset examples:

- Jointly incorporating the question and answer into the model input, which we refer to as the **qa** subset.
- Using only the full answer text, which we refer to as the **full** subset.
- Splitting the answer text into sentences, resulting in shorter text segments and producing 455,320 training examples and 114,117 validation examples. We refer to this subset as the **sentence** subset.

To test the robustness of our approach against adversarial attacks, we add misspellings and simulate homoglyph substitution on the **full** subset of the test sets, using the *nlaug* (Ma, 2019) library.

Regarding our choices of training hyperparameters, we maintain a fixed batch size of 32, adopt a linear scheduler with a warmup ratio of 0.1%, and restrict our learning rate tuning to a range between 10^{-5} and 5.10^{-5} with a step size of 10^{-5}. Our model is trained for 5 epochs, and we report the results averaged over 5 distinct random seeds for all in-domains results. For the out-of-domains experiments, we used the best models.

4.2 Results

4.2.1 In Domain

Table 1 presents the results obtained from hyperparameter tuning. Notably, both evaluated French models demonstrated exceptional performance, and consistent stability evidenced by the low standard deviation scores. However, the scores for French models were comparatively lower than the English models, indicating the impact of translation on model performance. Our findings suggest that the performance of models trained on the QA and Full subset significantly deteriorates when assessed on short-length or sentence data, indicated also by the high standard deviation scores. Conversely, models trained on sentences exhibit a relatively consistent performance across all subsets. Considering the overall highest performance on the **Full** subset, we opted to conduct subsequent experiments with the CamemBERTa and RoBERTa models trained on the Full subset.

Furthermore, Table 2 displays a detailed breakdown of the scores obtained from the **Full** subset. Notably, the models consistently achieve high recall scores in identifying ChatGPT across all tested languages. Additionally, the inclusion of misspellings and homoglyph substitutions improves the

[7]We also tested mDeBERTa (He *et al.*, 2021) but it wasn't converging in any of our hyper-parameter tuning experiments.

models' ability to detect human-written text while slightly reducing their performance for machine-generated text. The multilingual model XLM-R demonstrates superior and more resilient performance on both the French and English test sets, exhibiting increased robustness against adversarial attacks. When compared to a native French-speaking human linguist, the trained model accurately identifies ChatGPT-generated content with higher accuracy, while the human linguist achieves a similar human detection score.

Train	QA			Full			Sentence		
Test	QA	Full	Sentence	QA	Full	Sentence	QA	Full	Sentence
				French					
CamemBERT	**98.37±0.5**	97.79±0.4	**40.20±8.6**	**92.43±1.2**	**98.44±0.4**	25.08±4.7	**93.48±5.2**	96.41±0.6	90.27±0.3
CamemBERTa	98.23±0.3	**98.48±0.3**	32.00±6.3	90.13±1.0	**98.49±0.4**	**29.11±3.6**	81.82±3.4	**96.71±0.1**	**91.18±0.2**
				English					
RoBERTa	**99.88±0.03**	**98.91±0.2**	51.23±7.6	**98.58±0.7**	**99.86±0.03**	**66.93±5.4**	71.10±19.4	**99.39±0.07**	**98.17±0.1**
ELECTRA	99.27±0.2	**99.07±0.2**	**65.23±8.3**	96.24±0.7	99.35±0.1	43.82±9.1	**93.57±1.9**	97.05±0.4	93.60±0.1

Table 1: Average and standard deviation of F1 scores for the best model on the validation set with adversarial perturbations.

Evalution set		French			English			Multilingual						Human Expert		
								French-Test			English-Test					
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Full subset	ChatGPT	0.95	1	0.97	0.99	1	0.99	0.99	1	0.99	0.98	1	0.99	0.98	0.87	0.92
	Human	1	0.94	0.97	1	0.99	0.99	1	0.99	0.99	1	0.98	0.99	0.88	0.98	0.93
+misspelling	ChatGPT	1	0.95	0.98	0.99	0.79	0.88	1	0.96	0.98	0.99	0.99	0.99	-	-	-
	Human	0.95	1	0.98	0.82	0.99	0.9	0.96	1	0.98	0.99	0.99	0.99	-	-	-
+homoglyphs	ChatGPT	1	0.94	0.97	0.99	0.87	0.93	1	0.97	0.99	0.99	0.99	0.99	-	-	-
	Human	0.94	1	0.97	0.88	0.99	0.93	0.99	0.99	0.99	0.97	1	0.99	-	-	-

Table 2: Detailed test set scores (full subset) breakdown of CamemBERTa (French), RoBERTa (English), XLM-R (Multilingual) trained on the full subset.

4.2.2 Out-of-Domain

To assess the potential for overfitting to our in-domain data, we evaluated the performance of our French detector on the out-of-domain test sets described previously. The results, shown in Table 3, reveal the detector's exceptional performance on the FTB and FAQ-Gouv test sets, with a drop in accuracy to 88.75 on the FAQ-Rand subset. This suggests the model may be detecting translation artifacts that remain in some FAQ web pages after automatic translation. Remarkably, our detector correctly identified French text generated natively by ChatGPT, suggesting that it may be possible to develop detectors for other languages by translating existing datasets. Similarly, the detector models displayed surprising performance in detecting content generated by BingGPT.

The multilingual detector model consistently outperformed the monolingual model only in detecting human-generated text but fell behind in detecting ChatGPT or BingGPT-generated text, this behavior might be due to the significantly larger pre-training dataset of XLM-R compared to CamemBERTa.

The detector models also exhibited clear weaknesses against misspelling and homoglyph-based attacks. For instance, the performance of CamemBERTa and XLM-R dropped to 44.81 and 28.18, respectively, when detecting BingGPT-generated text with misspellings added.

Finally, the low scores obtained by the detector on the adversarial response dataset we developed in the style of ChatGPT and BingGPT serve to validate our detector's heavy reliance on the writing style utilized in generating responses.

True label	Human												ChatGPT					
Model	FTB			FAQ-Rand			FAQ-Gouv			Adversarial			Native			BingGPT		
	raw	*+ms*	*+hg*	raw	*+ms*	*+hg*	*raw*	*+ms*	*+hg*	*raw*	*+ms*	*+hg*	raw	*+ms*	*+hg*	raw	*+ms*	*+hg*
CamemBERTa	99.19	**99.92**	**100**	88.75	99.01	99.10	96.17	**100**	**99.57**	33.57	87.61	**85.49**	99.19	81.42	84.96	**92.45**	44.81	48.37
XLM-R	**99.43**	99.59	99.76	**95.35**	**99.39**	**99.55**	**96.59**	**100**	**99.57**	**59.12**	**89.05**	82.67	94.69	60.18	62.83	77.46	28.18	35.72
*Trained on a mix of raw, misspellings and homoglyphs**																		
CamemBERTa	98.98	98.54	98.79	80.56	84.51	84.73	90.64	91.49	90.21	45.90	42.62	44.26	**100**	**99.12**	**99.95**	91.51	**91.51**	**90.57**
XLM-R	98.54	98.78	98.79	85.20	88.84	95.32	92.34	96.17	95.32	62.26	60.66	62.30	100	97.34	99.16	62.26	53.77	56.60

Table 3: Accuracy scores of CamemBERTa and XLM-R on the French out-of-domain test sets. *ms*: misspelling, *hg*: homoglyphs. **Dataset mix was 100% raw, 50% misspellings and 50% homoglyphs.*

5 Discussion

About the link between translation quality and the model detectability As part of our study to assess the possibility of differentiating between texts written by humans and those generated by LLMs, following the work of Guo *et al.* (2023), we analyzed and re-evaluated the responses in the translated French dataset. The purpose was to confirm the hypothesis that a human expert can generally distinguish between a ChatGPT-generated text and one written by a human. We initially rated the translation quality on a scale of 1 to 5, with 5 indicating a good translation. Translations with scores exceeding 3 were retained even though ChatGPT managed to interpret badly translated questions extremely well.

Additionally, we assessed the correlation between our detector's performance and translation quality scores, and found it to be weak.[8]

About discriminating linguistics clues We identified several visible characteristics in the generated texts. ChatGPT uses an impersonal and didactic style, characterized by extensive use of the impersonal form, conditionals statements, as shown here:

- "Cela **pourrait** également nuire à la réputation de l'entreprise (...)"
- "Cela **pourrait** entraîner une baisse des dépenses des consommateurs (...)"
- "**Si** vous êtes allergique aux chats, cela signifie que votre corps a une réaction anormale aux protéines présentes dans leur peau, leur urine ou leur salive. **Si** vous deviez manger de la viande de chat, il est possible que (...)"

The language model structures its responses to create an impression of coherence and clarity. It often reformulates the question in its answer, resulting in a didactic response that aligns with the question.

[8]With three different correlations measures showing the same trend: Spearman's τ of -0.25, Pearson's R of -0.26 and Kendall's τ of -0.24.

- Question: “Pourquoi **mon signal wifi semble se dégrader avec le temps** ? Je réinitialise/redémarre constamment mon routeur et/ou mon modem. Je dois noter que je vis dans un petit appartement et que j’ai utilisé 2 routeurs haut de gamme. Explique comme si j’avais cinq ans.”
  GPT : “Il peut y avoir plusieurs raisons pour lesquelles **votre signal Wi-Fi se dégrade avec le temps**. Voici quelques explications possibles : (...)”

ChatGPT’s responses are often general, and it redefines the subject on which the question is asked.

When asked “How does nature solve for Pi ?” or “*Comment la nature résout-elle pour Pi ?”, it started by stating the definition of Pi:

- “Pi, ou le nombre 3,14, est une constante mathématique qui représente le rapport de la circonférence d’un cercle à son diamètre. La valeur de Pi est d’environ 3,14, mais c’est un nombre irrationnel (...)”

Additionally, ChatGPT is characterized by the absence of some human markers, such as errors in punctuation, spelling, or grammar. The language model does not use any tone, judgment, or personal touch, such as (“je pense que” / “je juge que”) , which creates a neutral impression. While its responses lack a human touch, it provides a specific recommendation when discussing technical or sensitive issues, such as consulting a specialist or seeking medical attention. Also, It does not ask any questions except towards the end of the response.

- “(...) **il est important de consulter un médecin** dès que possible. **N’hésitez pas à appeler le** 911 ou votre numéro d’urgence local si vous ressentez des douleurs à la poitrine ou d’autres symptômes d’une condition médicale grave.”
- “(...) **Il est important de** vérifier les instructions de votre four à micro-ondes pour voir si le support en métal peut être utilisé en toute sécurité.”
- “(...) Encore une fois, **je vous recommande de** parler avec un dermatologue ou un autre professionnel de la santé pour déterminer le plan de traitement le plus approprié pour votre cas spécifique.”
- “(...) J’espère que cela vous aidera à l’expliquer ! **Y a-t-il autre chose que vous aimeriez savoir sur Vénus ou sur la façon dont elle se déplace dans l’espace ?**”

Our study suggests that these visible differences could be used to differentiate between human-written texts and AI-generated texts automatically. It shall be noted that the ChatGPT tendency to produce didactic text can lead any detector trained on its content to be easily fooled assuming the text follows the same patterns. This is what showed our results in Table 3 (“Adversarial” column results).

About the character-level perturbations Interestingly, the introduction of character-level perturbations increased the model’s capability of detecting the adversarial human content, albeit at the expense of its capacity to detect Bing automatically generated content. This finding suggests that the addition of perturbations to content renders it more comparable to human-generated content, confirming, for French, previous work on the subject (Wolff & Wolff, 2020). These effects were much more difficult to notice in the in-domain scenarios because of the high-accuracy of the model.

Enhancing the robustness to noise of our models Although not the focus of this work, one obvious path of improvement is to add the same kind of perturbations to the training data in order to make the model more robust. To this end, we performed a quick set of experiments where we added to the training set, 50% of its content perturbed by mispellings and 50% with homoglyphs leading to a training set twice as big.[9] These results, presented in the lower half of Table 3, demonstrate that both models exhibit a minor decrease in human detection accuracy. However, they achieve substantial enhancements and improved robustness, particularly when utilizing CamemBERTa, for detecting ChatGPT-generated text in the presence of noisy data. Consequently, the detector models are now less inclined to attribute writing errors to human authors and instead focus more on writing style. This is evident from the scores obtained on the Adversarial set, where the performance on noisy data aligns more closely with that on the original set. However, this does not make the model less sensitive to other kinds of noises but it is an interesting path of improvement. As always with noisy adversarial user-generated content, the question is to find a more general approach that will avoid a constant *cat and mouse* game when it comes to processing productive content.

Take home message The key takeaway from our study is that detecting adversarial text, which is designed to evade detection by language models, presents a significant challenge. OpenAI has reported[10] a success rate of 26% in their own supervised settings when identifying adversarial content in a challenge set of English text.[11] Furthermore, OpenAI has stated that their detection methods are unreliable for text shorter than 1000 characters. We would like to emphasize that our study does not claim to have produced an universally accurate detector. Our strong results are based on in-domain testing and, unsurprisingly, do not generalize in out-of-domain scenarios. This is even more so when used on text specifically designed to fool language model detectors and on text intentionally stylistically similar to ChatGPT-generated text, especially instructional text.
We are currently extending the adversarial dataset using much more various sources as we believe that understanding the shortcoming of these models is of crucial importance.

6 Conclusion

In conclusion, this paper proposed a methodology for developing and evaluating ChatGPT detectors in multiple languages, focusing on French as a case study. The proposed method involved translating an English dataset into French and training a classifier on the translated data. The results demonstrate that the proposed method can effectively detect ChatGPT-generated text, with a certain degree of robustness against basic attack techniques, albeit exclusively within the in-domain setting. However, the detectors display evident vulnerabilities in out-of-domain contexts, emphasizing the importance of considering different writing styles in training language models. Additionally, the study highlights the significant challenge of detecting adversarial text, which even OpenAI's detection methods have difficulties with. The key takeaway is that caution should be exercised when applying in-domain testing results to a wider variety of content. We provide open-source resources to further advance research in this and are currently working to extend the adversarial dataset to better understand the limitations of these models.

[9] We also tested a 50% original training set + 25% misspelling + 25% homoglyphs pertubations model that led to slightly inferior performance, less than one percentage point of difference.

[10] https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

[11] Not released as the time of writing.

Acknowledgments

We thank the reviewers for their insightful comments. This work was partly funded by Benoît Sagot's chair in the PRAIRIE institute funded by the French national research agency (ANR as part of the "Investissements d'avenir" program under the reference ANR-19-P3IA-0001). This work also received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 101021607. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

ABEILLÉ A., CLÉMENT L. & KINYON A. (2000). Building a treebank for French. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece: European Language Resources Association (ELRA).

ANTOUN W., BALY F. & HAJJ H. (2021a). AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, p. 191–195, Kyiv, Ukraine (Virtual): Association for Computational Linguistics.

ANTOUN W., BALY F. & HAJJ H. (2021b). AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, p. 196–207, Kyiv, Ukraine (Virtual): Association for Computational Linguistics.

BENDER E. M., GEBRU T., MCMILLAN-MAJOR A. & SHMITCHELL S. (2021). On the dangers of stochastic parrots: Can language models be too big? DOI : 10.1145/3442188.3445922.

BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. *Advances in neural information processing systems*, **33**, 1877–1901.

CHEN S., JU Z., DONG X., FANG H., WANG S., YANG Y., ZENG J., ZHANG R., ZHANG R., ZHOU M., ZHU P. & XIE P. (2020). Meddialog: a large-scale medical dialogue dataset. *arXiv preprint arXiv:2004.03329*.

CHOWDHERY A., NARANG S., DEVLIN J., BOSMA M., MISHRA G., ROBERTS A., BARHAM P., CHUNG H. W., SUTTON C., GEHRMANN S. *et al.* (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

CHRISTIANO P. F., LEIKE J., BROWN T., MARTIC M., LEGG S. & AMODEI D. (2017). Deep reinforcement learning from human preferences. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 30: Curran Associates, Inc.

CLARK K., LUONG M.-T., LE Q. V. & MANNING C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, p. 8440–8451, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.747.

DE BRUYN M., LOTFI E., BUHMANN J. & DAELEMANS W. (2021). MFAQ: a multilingual FAQ dataset. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, p. 1–13, Punta Cana, Dominican Republic: Association for Computational Linguistics. DOI : 10.18653/v1/2021.mrqa-1.1.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota: Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

FAGNI T., FALCHI F., GAMBINI M., MARTELLA A. & TESCONI M. (2021). Tweepfake: About detecting deepfake tweets. *Plos one*, **16**(5), e0251415.

FAN A., JERNITE Y., PEREZ E., GRANGIER D., WESTON J. & AULI M. (2019). ELI5: long form question answering. In A. KORHONEN, D. R. TRAUM & L. MÀRQUEZ, Éds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, p. 3558–3567: Association for Computational Linguistics. DOI : 10.18653/v1/p19-1346.

FEDUS W., ZOPH B. & SHAZEER N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

GUO B., ZHANG X., WANG Z., JIANG M., NIE J., DING Y., YUE J. & WU Y. (2023). How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

HE P., GAO J. & CHEN W. (2021). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

HOFFMANN J., BORGEAUD S., MENSCH A., BUCHATSKAYA E., CAI T., RUTHERFORD E., CASAS D. D. L., HENDRICKS L. A., WELBL J., CLARK A. *et al.* (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

JAWAHAR G., ABDUL-MAGEED M. & LAKSHMANAN L. (2022). Automatic detection of entity-manipulated text using factual knowledge. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 86–93, Dublin, Ireland: Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-short.10.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

MA E. (2019). Nlp augmentation. https://github.com/makcedward/nlpaug.

MAIA M., HANDSCHUH S., FREITAS A., DAVIS B., MCDERMOTT R., ZARROUK M. & BALAHUR A. (2018). Www'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of The Web Conference 2018*, WWW '18, p. 1941–1942, Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. DOI : 10.1145/3184558.3192301.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

MCGUFFIE K. & NEWHOUSE A. (2020). The radicalization risks of gpt-3 and advanced neural language models. *arXiv preprint arXiv:2009.06807*.

MITCHELL E., LEE Y., KHAZATSKY A., MANNING C. D. & FINN C. (2023). Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

NGUYEN-SON H.-Q., THAO T., HIDANO S., GUPTA I. & KIYOMOTO S. (2021). Machine translated text detection through text similarity with round-trip translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 5792–5797.

OUYANG L., WU J., JIANG X., ALMEIDA D., WAINWRIGHT C., MISHKIN P., ZHANG C., AGARWAL S., SLAMA K., GRAY A., SCHULMAN J., HILTON J., KELTON F., MILLER L., SIMENS M., ASKELL A., WELINDER P., CHRISTIANO P., LEIKE J. & LOWE R. (2022). Training language models to follow instructions with human feedback. In A. H. OH, A. AGARWAL, D. BELGRAVE & K. CHO, Éds., *Advances in Neural Information Processing Systems*.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). Improving language understanding by generative pre-training.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

RAE J. W., BORGEAUD S., CAI T., MILLICAN K., HOFFMANN J., SONG F., ASLANIDES J., HENDERSON S., RING R., YOUNG S. *et al.* (2021). Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, **21**(140), 1–67.

SHOEYBI M., PATWARY M., PURI R., LEGRESLEY P., CASPER J. & CATANZARO B. (2019). Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

SOLAIMAN I., BRUNDAGE M., CLARK J., ASKELL A., HERBERT-VOSS A., WU J., RADFORD A., KRUEGER G., KIM J. W., KREPS S. *et al.* (2019). Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

STIENNON N., OUYANG L., WU J., ZIEGLER D., LOWE R., VOSS C., RADFORD A., AMODEI D. & CHRISTIANO P. F. (2020). Learning to summarize with human feedback. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems*, volume 33, p. 3008–3021: Curran Associates, Inc.

UCHENDU A., LE T., SHU K. & LEE D. (2020). Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 8384–8395, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.673.

WEIDINGER L., MELLOR J., RAUH M., GRIFFIN C., UESATO J., HUANG P.-S., CHENG M., GLAESE M., BALLE B., KASIRZADEH A. *et al.* (2021). Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

WOLFF M. & WOLFF S. (2020). Attacking neural text detectors. *arXiv preprint arXiv:2002.11768*.

YANG Y., YIH S. W.-T. & MEEK C. (2015). Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*: ACL - Association for Computational Linguistics.

ZELLERS R., HOLTZMAN A., RASHKIN H., BISK Y., FARHADI A., ROESNER F. & CHOI Y. (2019). Defending against neural fake news. *Advances in neural information processing systems*, **32**.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Cross-lingual Strategies for Low-resource Language Modeling: A Study on Five Indic Dialects

Niyati Bafna[1] Cristina España-Bonet[3] Josef van Genabith[2,3]
Benoît Sagot[1] Rachel Bawden[1]
(1) Inria, Paris, France
(2) Saarland Informatics Campus, Saarland University, Germany
(3) DFKI GmbH, Germany
niyatibafna13@gmail.com,{josef.van_genabith,cristinae}@dfki.de,
{benoit.sagot,rachel.bawden}@inria.fr

ABSTRACT

Neural language models play an increasingly central role for language processing, given their success for a range of NLP tasks. In this study, we compare some canonical strategies in language modeling for low-resource scenarios, evaluating all models by their (finetuned) performance on a POS-tagging downstream task. We work with five (extremely) low-resource dialects from the Indic dialect continuum (Braj, Awadhi, Bhojpuri, Magahi, Maithili), which are closely related to each other and the standard mid-resource dialect, Hindi. The strategies we evaluate broadly include from-scratch pretraining, and cross-lingual transfer between the dialects as well as from different kinds of off-the-shelf multilingual models; we find that a model pretrained on other mid-resource Indic dialects and languages, with extended pretraining on target dialect data, consistently outperforms other models. We interpret our results in terms of dataset sizes, phylogenetic relationships, and corpus statistics, as well as particularities of this linguistic system.

RÉSUMÉ

Stratégies inter-langues pour la modélisation des langues à faibles ressources : étude sur cinq dialectes indo-aryens

Les modèles de langue neuronaux jouent désormais un rôle central en traitement automatique des langues, grâce à leurs performances sur de nombreuses tâches du domaine. Dans cet article, nous étudions le développement de tels modèles pour des langues (très) peu dotées, à savoir cinq langues du continuum dialectal indo-aryen (braj, awadhi, bhojpuri, magahi, maithili), toutes très proches du hindi, une langue moyennement dotée. Nous comparons plusieurs stratégies classiques par l'adaptation (*finetuning*) et l'évaluation sur la tâche d'étiquetage en parties du discours. Ces stratégies incluent le pré-entraînement à partir de zéro ainsi que le transfert entre dialectes et depuis des modèles multilingues existants. Nous constatons qu'un modèle préentraîné sur d'autres dialectes et langues indiennes moyennement dotées avec poursuite du préentraînement sur les données du dialecte cible surpasse systématiquement les autres modèles. Nous interprétons nos résultats à la lumière de la taille des jeux de données et de leurs propriétés statistiques, des relations phylogénétiques entre dialectes, ainsi que des particularités de ce système linguistique.

KEYWORDS: Language modeling, low-resource, Indic languages, cross-lingual transfer, POS tagging.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 28–42.

MOTS-CLÉS : Modèles de langue, peu dotée, langues indo-aryennes, transfert cross-lingue, étiquetage morpho-syntaxique.

1 Introduction

In the last decade, natural language models have made tremendous progress on multiple tasks (Kalyan *et al.*, 2021). Many recent advances in natural language processing (NLP) owe credit to neural models that are pretrained over large quantities of unlabeled text, such as BERT (Devlin *et al.*, 2019). Data inequity over the vast range of the world's languages has led efforts to "transfer" these data-hungry neural models from resource-rich languages such as English and Spanish, to lower-resource languages (Wu & Dredze, 2019), with many studies focusing on phylogenetic closeness between the source and target languages as one of the important factors determining the results (Lin *et al.*, 2019; Dhamecha *et al.*, 2021; Patil *et al.*, 2022).

Indic NLP, or NLP for Indian languages,[1] has also made corresponding advances, with the release of large corpora, language models, and benchmarks for 18 "major" Indian languages (Kakwani *et al.*, 2020). However, there are hundreds of other languages and dialects in India, many of them spoken by millions of people, such as Rajasthani, Kannauji, Garhwali, and others, that have non-existent or nascent NLP research (Bafna *et al.*, 2022).

We work with a typical real-world situation, with five (extremely) low-resource North Indian dialects belonging to the Indic language family—namely, Braj, Awadhi, Bhojpuri, Magahi, Maithili. These languages bear close relationships with Hindi, a mid-resource dialect[2] although with a number of morphosyntactic and lexical divergences. We compare strategies for language modeling for low-resource languages, using a part-of-speech (POS) tagging downstream task for evaluation. We report that relatedness at the level of the language family between the pretraining languages and the target language benefits downstream performance. However, while comparing low-resource dialects as sources for a particular target dialect in a transfer setup, the results are less explained by phylogeny than by corpus domain match and lexical overlap. Finally, we find that extended pretraining shows consistent benefits. We hope that these experiments will raise an interest in NLP for these dialects, and constitute a starting point for other work in this context.

2 Related Work

Recently, there have been some attempts to develop basic NLP tools and resources for some of the prominent languages of the Indic continuum. Mundotiya *et al.* (2021) collect monolingual data and POS-tagged corpora for Bhojpuri, Maithili, and Magahi, also providing CRF baselines for POS tagging. Priyadarshi & Saha (2020) collect a monolingual corpus for Maithili as well as some POS-annotated data,[3] Ojha (2019) contribute a similar effort for Bhojpuri, and Ojha *et al.* (2020)

[1]The word "Indic", depending on context, is used to refer to both the subfamily of the Indo-European family spoken in India (e.g. Hindi, Bengali, Marathi), and the Indian languages in general (including non-Indo-European languages such as the Dravidian family of languages). In this paper, unless otherwise mentioned, we use the first, phylogenetic, sense of the term.

[2]In this work, we will refer to Braj (bra), Awadhi (awa), Bhojpuri (bho), Magahi (mag), Maithili (mai), and (standard) Hindi as "dialects" belonging to the "macrolanguage" of the dialect continuum that forms the "Hindi" heartland of India. This terminology is not intended to have political connotations.

[3]Not publicly available.

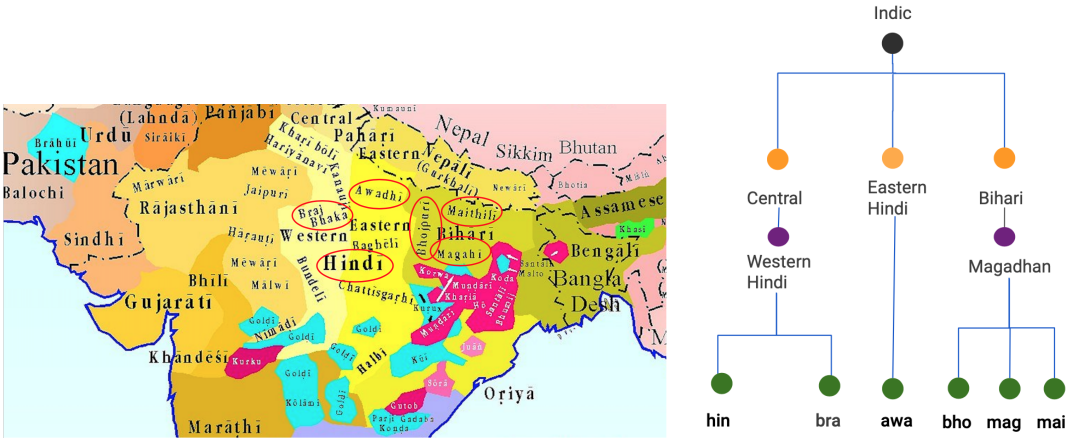
Hindi	Awadi	Bhojpuri	Magahi	Maithili	Meaning
dʑaː rəheː hoː	dʑaːt̪ əhaːi	dʑaːt̪ baː	dʑaː həi	dʑaː rəhəl əʧʰ i	(you) are going
ləɖkɑː	ləɖkɑː	ləikaː	ləiːkaː	ləɖkɑː	boy (nom.)
bət̪aːjaː/ kəːh lijaː	bət̪aːvət̪	kəhəl	kəhəlieː	kəhəlhuⁿ	told (completive)
aːpkiː	aːpən	aːpən	əpən	əhaːnk	your (hon., fem. sing. obj)
bəhən	bəhin	bəhin	bəhin	bəhin	sister

Table 1: Examples of cognates. Braj is not included due to lack of data. Since the Devanagari script is phonetically transparent, phonetic similarity is visible both in IPA and in Devanagari (not shown).

provide monolingual data and some parallel data for Bhojpuri and Magahi. As part of the NSURL 2019 shared task in POS-tagging for Bhojpuri and Magahi (Freihat & Abbas, 2019), Kumar M (2019) present an SVM-based system as well as a BERT-based classifier. Proisl *et al.* (2019) experiment with available taggers, including a BiLSTM+CRF architecture and the Stanford tagger.

There has also been work in language modeling for "dialects" of a standard variant, accompanying, of course, a rich literature in cross-lingual transfer to low-resource languages. Transformer-based pre-trained multilingual models such as mBERT (Devlin *et al.*, 2019; Conneau *et al.*, 2020) are often claimed to show multilingual generalization (Pires *et al.*, 2019). There are multiple aspects to the phenomenon of multilingual generalization, and many of them have received attention in the NLP community. One of the primary ways in which cross-lingual ability is demonstrated is through zero-shot transfer, i.e. a setting in which a multilingual pretrained model is trained on labeled or supervised data in one language and performs well in another language. Early papers found that mBERT performed remarkably well in the zero-shot setting (Pires *et al.*, 2019; Wu & Dredze, 2019) under certain conditions, such as similar typologies of source and target languages, but regardless of others, such as script and common vocabulary. Since then, many studies, such as (Chai *et al.*, 2022; Ri & Tsuruoka, 2022), have attempted to explore these conditions; notably, Muller *et al.* (2021) show that a common script indeed facilitates transfer, along with shared typological features, and Khemchandani *et al.* (2021) show the same for Indic languages. Studies are split on results regarding the relationship between subword overlap and ease of transfer. For example, K *et al.* (2020) show that shared subwords play a small role in positive transfer, while Deshpande *et al.* (2022) argue that this is only the case for languages with shared word order.

Research has also looked at the question of which languages may benefit from large multilingual models. For high-resource languages, it was quickly clear that monolingual models outperform or at least match multilingual models on most tasks (de Vries *et al.*, 2019; Martin *et al.*, 2020). However, ensuing studies have also found that monolingual models or language-family models can outperform multilingual counterparts for low-resource languages (Ulčar & Robnik-Šikonja, 2020; Ortiz Suárez *et al.*, 2020; Armengol-Estapé *et al.*, 2021; Micallef *et al.*, 2022; Barry *et al.*, 2022). In effect, there is no clear consensus in the community on the best strategies for solving a downstream task in typical conditions for a low-resource language, specifically, limited monolingual data, a related high-resource language, and possibly some annotated task data. This motivates our work in investigating different cross-lingual transfer strategies in the given context of dialects from the Indic continuum.



(a) Geography of the Indic dialect contiuum (b) Theoretical phylogenetic tree.

Figure 1: Characterization of the Indic dialects used in this work

3 Languages

The Indic dialect continuum is a group of more than 40 dialects spoken across most of North India and surrounding regions, with hundreds of millions of speakers. The dialects of this continuum are classified under the Apabhramsic (e.g. Rajasthani), Western Hindi (e.g. Haryanvi), Eastern Hindi (e.g. Awadhi), and Bihari (e.g. Bhojpuri) branches of the Indic language family. We are working with six of these dialects, all of which are written in the Devanagari script, namely Awadhi, Bhojpuri, Braj, Magahi, Maithili, and the high-resource standardized dialect, i.e. Hindi. See Figure 1b for a phylogenetic tree of these dialects.[4] Geographically, these dialects are spread across the continuum (see Figure 1a[5]), with standard Hindi co-existing in many of these regions, although it is genetically part of the western sub-family of the continuum. This means that many of these dialects borrow from Hindi and share similarities with it due to contact, rather than via genetic transmission.

The dialects of the Indic continuum share cognates as well as morphosyntactic properties, such as a roughly common (free) word order, noun inflection for case, and verbal inflection for number and gender to a varying extent. However, they differ in specifics, for example, in the number of cases, the levels of honorifics, and the degree of inflection for gender. See Table 1 for examples of cognates in these dialects.[6]

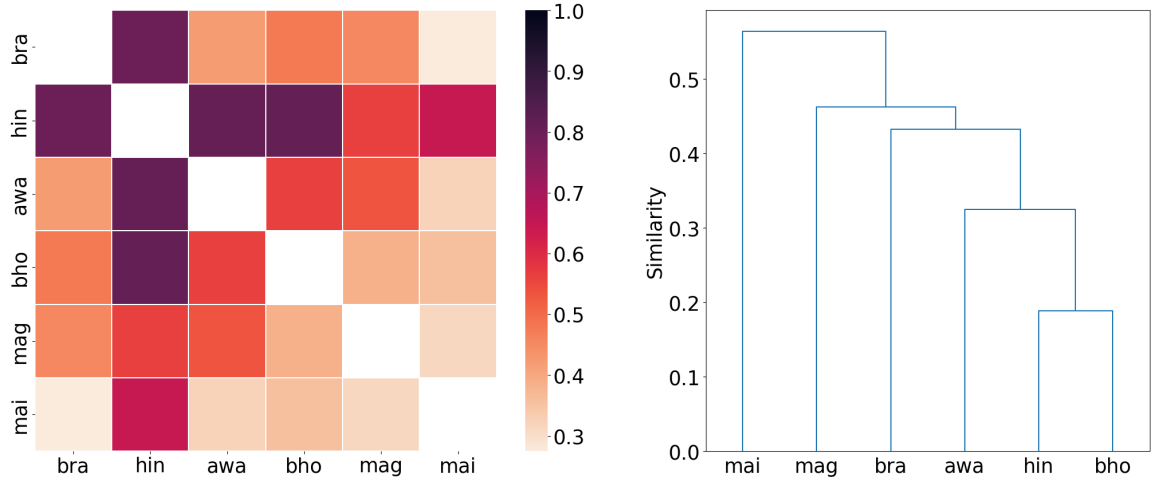
4 Data and Description

Data We use monolingual data from different (non-overlapping) sources for these dialects. These sources include the VarDial 2018 shared task (Zampieri *et al.*, 2018) for Bhojpuri, Magahi, Awadhi, and Braj, the BHLTR project for Bhojpuri (Ojha, 2019), LoResMT (Ojha *et al.*, 2020) for Bhojpuri and Magahi, and the BMM corpus (Mundotiya *et al.*, 2021) and the Wordschatz Leipzig corpus (Goldhahn *et al.*, 2012) for Maithili. For Hindi, we use the IndicCorp corpus (Kakwani *et al.*, 2020). This is the largest available consolidated Hindi corpus, as of the date of writing, and was used to

[4]Taken from Glottolog: https://glottolog.org/resource/languoid/id/midd1375.
[5]Taken from https://titus.fkidg1.uni-frankfurt.de/indexe.htm
[6]Translations are taken from Glosbe: https://glosbe.com.

	Monolingual #toks	POS #toks	POS labels #tags
awa	0.16M	21K	37
bho	2.99M	94K	34
bra	0.32M	62K	31
hin	1800.00M	351K	31
mag	3.04M	61K	19
mai	0.46M	211K	25

Table 2: Monolingual/POS dataset sizes (in #tokens) and POS tagset sizes, for all dialects.



(a) Pairwise lexical similarity for all dialects.

(b) Dendrogram, from pairwise similarities

Figure 2: Visualizations of lexical similarity among dialects.

train the publicly available pretrained Hindi model that we use for our experiments (described in Section 5).

For POS-annotated datasets, we use a single data source in a given dialect to maintain consistency in annotation style, the tagset, and its associated granularity, although these differ across dialect datasets. Specifically, we use the NSURL 2019 shared task datasets (Freihat & Abbas, 2019) for Bhojpuri and Magahi, the KMI-Linguistics datasets[7] for Awadhi and Braj, the BMM corpus for Maithili, and the Universal Dependencies Treebank HDTB project for Hindi (Palmer *et al.*, 2009; Bhat *et al.*, 2017). Aggregate token counts for each dialect are listed in Table 2.

Crosslingual interaction We would like to understand the crosslingual interactions between these dialects, in order to contextualize the results of our experiments, described in Section 5. We calculate the normalized lexical overlap[8] between monolingual corpora for all pairs of dialects (Figure 2a). The resulting similarity matrix can be used to cluster the dialects by similarity (as shown in Figure 2b); we see that the resulting tree does not resemble the gold phylogenetic tree in Figure 1b. This can have a few explanations: it is possible that lexical similarity is not a good measure of closeness—it does

[7] https://github.com/kmi-linguistics/

[8] This is calculated as the number of unique words that are common to both corpora, divided by the minimum of the number of unique words in the two corpora; thus, the measure lies between 0 and 1.

not, for example, take into account morphosyntactic similarity, or shared cognates that do not exactly match. It is also possible that the corpora we use are not representative of the true distributions of the dialects. Note that the outermost leaf in the tree, Maithili, is the only dialect whose monolingual data does not include VarDial shared task data. Finally, we can also credit widespread borrowing of words from Hindi with genetically more distant dialects having higher-than-expected lexical similarity.

The lexical similarity between monolingual pretraining corpora and the annotated datasets[9] may also be relevant in explaining our transfer results (see Figure 3a). Intuitively, low lexical similarity would indicate fewer benefits of pretraining. We note that the Hindi corpus has an almost perfect coverage of the annotated datasets of all dialects including itself. This is unsurprising given its size as well as the common subsumption of dialects on this continuum under "Hindi."

5 Experiments

We compare different multilingual or cross-lingual transfer strategies in the context of Indic dialects, evaluated on the downstream task of POS-tagging. We will refer to the two following operations: "pretraining" refers to training a model on mono/multilingual raw text, "EP" or "extended pretraining" refers to further pretraining an already pretrained model on different mono/multilingual raw text.[10] All models are finetuned and evaluated on the annotated dataset of the target dialect. Due to tagset differences, we cannot attempt zero-shot transfer without performing label alignment. Finally, we do not pretrain from scratch on Hindi data and use a publicly available Hindi pretrained model instead (Joshi, 2022).

We use the HuggingFace transformers library (Wolf *et al.*, 2020) for training and accessing publicly available pretrained models. All models (including the publicly available models that we use) have a BERT-base architecture, consisting of 12 attention heads, 12 layers, and with hidden layer size 768. Models trained from scratch on dialect data are trained on the Masked Language Modeling (MLM) objective for ~40 epochs, EP over pretrained models is performed for 15 epochs, and finetuning is performed over the best performing pretraining checkpoint for 5 epochs.[11]

Baseline As the baseline for each dialect, we use the BERT architecture described above, pretrained from scratch on monolingual dialect data, and finetuned on task data for the dialect.

Using pretrained models We report the results obtained by finetuning three publicly available large pretrained models on the POS-tagged dataset for each dialect:

- Hindi (we hereafter refer to this model as Hin-BERT) (Joshi, 2022): We use a pretrained Hindi model[12] trained on the MLM objective; we want to see how well a pretrained model in a *related mid-resource dialect* transfers to a low-resource dialect.

[9] Calculated in the same way as for corpus lexical similarity.

[10] Other works may use different terms for what we call extended pretraining, or may carry out extended pretraining in a different manner.

[11] Further finetuning did not improve performance.

[12] https://huggingface.co/l3cube-pune/hindi-bert-scratch/tree/main

- MuRIL – Multilingual Representations for Indian Languages (Khanuja *et al.*, 2021): The publicly available MuRIL model represents a *mid/high-resource related language family* model. MuRIL is trained on the MLM and Translation Language Modeling (TLM) objectives on 17 languages in total, including other Indic languages such as Marathi and Bengali, as well as genealogically unrelated languages from the Indian subcontinent, such as Tamil and Kannada.[13]

- mBERT (Devlin *et al.*, 2019):[14] Finally, we also finetune mBERT for each dialect; mBERT is trained on the MLM and Next Sentence Prediction (NSP) objectives, on 104 languages, including Indic languages, but also several other languages and language families.

We also perform extended pretraining for the MuRIL and Hin-BERT models to observe potential benefits. Specifically, these models are pretrained further with an MLM objective on the monolingual data of the target dialect, and then (like all other models) finetuned and evaluated on the target dialect. The MuRIL model was chosen over mBERT due to its better initial performance.[15]

Using related dialects One can use data in related low-resource dialects to boost the learning of shared properties and similar words. We conduct the following experiments to investigate this idea:

- Pairwise transfer (D+ft): We pretrain a BERT model from scratch on monolingual data from a low-resource source dialect, therefore excluding Hindi, and finetune and evaluate it on task-specific data of the target dialect. We do this for all possible pairs, and report the best F1 performance over all source languages for every target dialect. We would also like to draw inference from the above setup as to which dialects perform best as sources for a given dialect, in relation to their genealogical or other type of closeness to the target. In the D+ft setup, however, the results are confounded by varying amounts of monolingual data available for different dialects. Therefore, we fix the monolingual as well as the evaluation data size in tokens for all (source and target) dialects, using the minimum available dataset size (Awadhi), and repeat pairwise transfer experiments.

- All dialects together (ABBMM+ft): In the setup, we pretrain a BERT model from scratch on all available low-resource dialect data,[16] and finetune and evaluate separately on each dialect. The aim of this experiment is to investigate how far joint training on related dialects (without a high-resource dialect) can benefit the target.

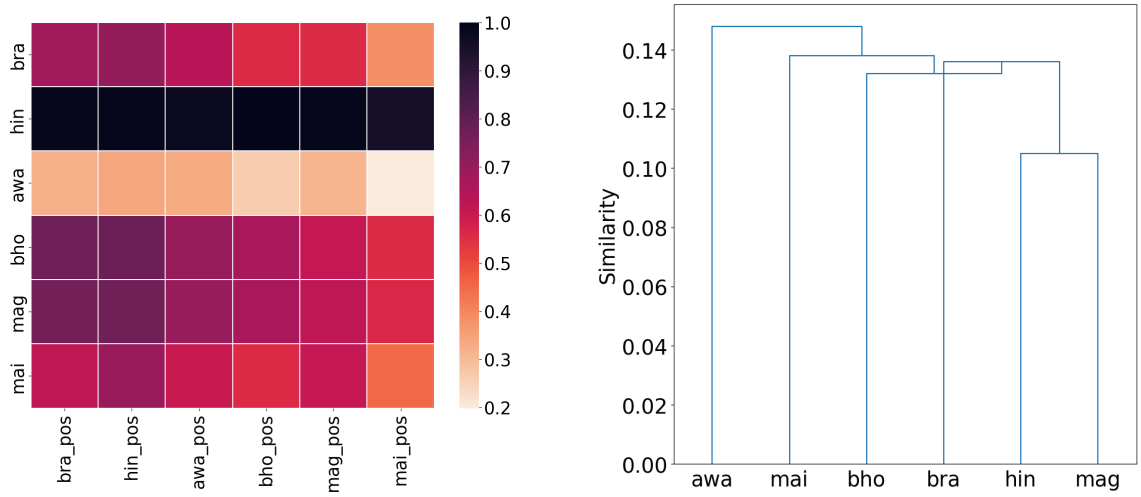
- MuRIL with EP on all dialects (MuRIL+EP$_{ABBMM}$+ft): Finally, we choose the best performing large pretrained model from the previous set of experiments (namely, MuRIL), and extended-pretrain it with all low-resource dialects, followed by separate finetuning and evaluation in each dialect.

[13] See more details here: https://huggingface.co/google/muril-base-cased/tree/main.

[14] https://huggingface.co/bert-base-multilingual-cased/tree/main

[15] We do not perform this experiment for Hindi, i.e. we do not do extended pretraining on Hindi data, for two reasons: firstly, both MuRIL and mBERT have already seen Hindi data, therefore rendering this a different experiment to that with the dialects, and secondly, in our work, our focus is on the low-resource dialects. We leave the exploration of best-performing transfer setups for mid-to-high range resource languages to other works.

[16] Hindi is not included in these experiments since it would easily dominate the low-resource data, and the resulting experiment would not be very different from Hin-BERT+ft, which we conduct separately.



(a) Lexical similarity of monolingual corpus with POS dataset (mono-POS overlap)

(b) Dendrogram from pairwise transfer results with fixed dataset sizes

Figure 3: Pairwise transfer results

	bho	bra	awa	hin	mag	mai
Monolingual	92.97	87.98	80.46	97.10	90.53	85.00
Hin-BERT+ft	92.13	93.31	84.05	97.10	90.89	87.98
Hin-BERT+EP$_{mono}$+ft	92.42	93.74	84.10	-	91.12	87.46
mBERT+ft	93.05	93.5	83.53	97.08	90.47	87.90
MuRIL+ft	92.96	94.14	82.31	98.01	91.25	88.54
MuRIL+EP$_{mono}$+ft	**93.62**	**94.73**	84.22	-	91.81	88.30
D+ft	91.40	92.12	82.95	96.39	90.39	87.01
ABBMM+ft	92.86	93.14	83.12	96.35	90.96	87.48
MuRIL+EP$_{ABBMM}$+ft	93.59	94.68	**85.72**	-	**91.95**	**88.69**

Table 3: Evaluation on POS-tagging (F1). EP: extended pretraining, ABBMM: all low-resource dialect data, D+ft: best performing low-resource source-to-target model per target dialect.

Target Lang	Source Language bho	bra	awa	hin	mag	mai
bho	**92.97**	87.49	87.67	92.13	91.40	87.48
bra	92.12	87.98	87.79	**93.31**	91.31	89.31
awa	82.95	80.06	80.46	**84.05**	82.43	79.88
hin	96.39	94.69	94.55	**97.10**	96.11	94.40
mag	90.39	87.97	88.06	**90.89**	90.53	87.72
mai	87.01	85.53	85.25	**87.98**	86.79	85.01

(a) Original dataset sizes

Source Language bho	bra	awa	hin	mag	mai
84.17	83.77	83.50	83.63	82.51	83.00
90.57	**93.31**	92.83	90.44	92.79	92.09
79.25	77.92	77.65	77.37	79.53	**80.19**
84.28	83.69	**87.02**	86.24	83.45	86.39
85.24	86.23	85.93	83.31	**86.87**	86.08
81.84	80.22	80.89	80.56	79.50	81.39

(b) Equal dataset sizes

Table 4: Evaluation on POS-tagging (F1), for pairwise transfer experiments.

6 Results and Discussion

Monolingual Performance Within dialects, differences in performance can be explained by the size of the annotated dataset, the amount of monolingual data, and the complexity of the tagset (see Table 2), as well as the token coverage of the annotated dataset by the monolingual data (mono-POS overlap, see Figure 3a). We see in Table 3 that the Maithili and Awadhi monolingual models perform worse than the others; they also show the least mono-POS overlap. Similarly, Braj has less monolingual and annotated data than Maithili, but still performs better, possibly because of its higher mono-POS overlap.

Bhojpuri and Magahi, the highest resourced dialects, have baseline performances roughly on par with {mBERT | MuRIL}+ft. Although these dialects are still very low-resource by several orders of magnitude compared to Hindi, their datasets are already big "enough" to yield a good performance on a relatively shallow task such POS-tagging. The transfer methods are mainly beneficial for the lowest-resourced dialects, Braj, Awadhi, and Maithili.

Pretrained models Comparing {Hin-BERT | MuRIL | mBERT}+ft, we observe that MuRIL-based models do better than mBERT-based models on four out of six dialects. This extends the results shown by Khanuja *et al.* (2021) that demonstrate that MuRIL outperforms mBERT consistently on its 17 pretraining languages. We also see that Hin-BERT and mBERT seem to perform on par, with perhaps a slight edge to Hin-BERT, although mBERT is pretrained on much more data. This corroborates the intuition that the relatedness of the pretraining languages with target languages could positively affect transfer results, "compensating", in a way, for less data. However, the fact that these pretrained models differ in their pretraining objectives must be kept in mind while making observations about the effects of pretraining languages.[17]

The increase in performance over the monolingual baseline with pretrained models, especially for Hin-BERT+ft, can also be contextualized by crosslingual lexical similarity between monolingual corpora (see Figure 2a). We see that Braj, Awadhi, and Bhojpuri show the highest lexical similarity with Hindi. This explains the jump in performance for low-resource Braj and Awadhi (whereas for Bhojpuri, which already has a good amount of monolingual data, training on Hindi causes worsening).

Extended pretraining EP helps consistently; language-specific pretraining possibly serves to expose the model to non-cognate words or language specific constructions in the target language. The only performance drop is observed for Maithili; monolingual EP slightly worsens performance in both Hin-BERT+ft and MuRIL+ft. This accords with our earlier observation of the low Maithili mono-POS overlap, and its possible effects.

Using other dialect data The best performing single-language transfer from another low-resource dialect, i.e. the D+ft model, does better than the monolingual model for dialects with little monolingual data, namely, Braj, Maithili and Awadhi, and worse for the higher resourced dialects i.e. Bhojpuri and Magahi. ABBMM+ft always does better than D+ft, presumably because the model sees monolingual data in the target dialect as well as more related dialect data in general. We also observe that

[17] Note that the training data of mBERT does include Hindi and other Indic languages; however, these are naturally accorded less "space" or percentage of training data as compared to with MuRIL or simply a Hindi pretrained model.

ABBMM+ft is on par with {mBERT | MuRIL}+ft even for dialects without much monolingual data; again, this indicates that models pretrained on roughly of a few million tokens, even from closely related dialects, perform comparably with much larger (language family or other) multilingual models (pretrained on three orders of magnitude more data) for a downstream POS-tagging task. It is possible that this is not the case for tasks requiring more language understanding.

Pairwise transfer with equal dataset size By fixing dataset sizes for all dialects, we aim to directly compare different dialects as (pretraining) sources for a given target dialect.[18] The resulting F1 scores are presented in Table 4a. We see that the differences in performance for a given dialect with different pretraining dialects are much lower than before. Interestingly, we observe that for Awadhi, Hindi, and Maithili, it is better to pretrain on a different dialect than itself. This can be partially understood in view of similarities between different dialect corpora and annotated datasets (Figure 3a), although these similarities are calculated over the full datasets rather than same-sized subsets. For example, we see that the Maithili POS-dataset has lower lexical overlap with the Maithili corpus than with the Bhojpuri corpus. A similar argument holds for Awadhi.

We also use these scores (Table 4b) to extract a dendrogram of language relatedness, with the hypothesis that this may recover a phylogenetic tree, which would mean that genetically close dialects behave similarly as sources and targets. We use the 0-1 normalized mean source-target performance of each pair of dialects as their "similarity" score[19] (Figure 3b). The resulting tree is not in fact a good representation of the phylogenetic tree of these dialects; the effect of genealogy seems to be outweighed by other factors, possibly including lexical overlap due to borrowings, and domain match.

Takeaways The takeaways from the results can be summarized as follows:

- Multilingual models pretrained on (a) data from the same language family, (b) a closely related high-resource dialect, (c) "general" multilingual data, as well as (d) low-resource closely related dialects are all good candidates for base pretrained models (to be finetuned on task data), with (a) consistently outperforming the others. Their relative performance can be interpreted as a function of the relatedness of the pretraining corpus languages, and the amount of such data.

- Among closely related dialects, the best performing source dialect pretrained model may be determined by lexical overlap or domain match with the target dialect annotated data rather than phylogenetic closeness between the source and target dialects; in general, especially for lower-resource dialects, closely-related dialect data helps performance.

- Extended pretraining, even on very little data, consistently helps. The best performing models are obtained by extended pretraining MuRIL with either monolingual data (for Bhojpuri and Braj) or all dialect data together (for Awadhi, Maithili, and Magahi).

[18] Although we do also fix the annotated dataset size, this does not mean that the downstream task is of the same difficulty for all dialects. Different datasets have different inherent difficulty due to the tagset size, length of sentences, rare words, tag distributions, etc. Therefore, it is still not advisable to make comparisons across target dialects.

[19] This clustering algorithm makes the assumption that the similarity of a dialect with itself is 1, or at least higher than that with any other dialect; we therefore ignore self-source-target scores.

7 Conclusion

In this paper, we looked at different strategies for developing language models for low-resource languages, using five extremely low-resource dialects belonging to the Indic continuum as a testbed. We compared conventional pretraining and cross-lingual transfer methods, and concluded that large pretrained models trained on the same language family (in our case MuRIL, for the Indic language family) are particularly successful as base models, especially if followed by extended pretraining, either monolingually or on closely related dialect data. We hope that this work contributes to building a basic research base for the Indic dialect continuum, as well as other dialect systems.

Acknowledgments

This work was partly funded by R. Bawden's and B. Sagot's chairs in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001 and by the Emergence project, DadaNMT, funded by Sorbonne Université. The work was also supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) under grant SFB 1102: Information Density and Linguistic Encoding.

References

ARMENGOL-ESTAPÉ J., CARRINO C. P., RODRIGUEZ-PENAGOS C., DE GIBERT BONET O., ARMENTANO-OLLER C., GONZÁLEZ-AGIRRE A., MELERO M. & VILLEGAS M. (2021). Are Multilingual Models the Best Choice for Moderately Under-resourced Languages? A Comprehensive Assessment for Catalan. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, p. 4933–4946.

BAFNA N., VAN GENABITH J., ESPAÑA-BONET C. & ŽABOKRTSKÝ Z. (2022). Combining Noisy Semantic Signals with Orthographic Cues: Cognate Induction for the Indic Dialect Continuum. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, p. 110–131, Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics.

BARRY J., WAGNER J., CASSIDY L., COWAP A., LYNN T., WALSH A., Ó MEACHAIR M. J. & FOSTER J. (2022). gaBERT — an Irish Language Model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 4774–4788, Marseille, France: European Language Resources Association.

BHAT R. A., BHATT R., FARUDI A., KLASSEN P., NARASIMHAN B., PALMER M., RAMBOW O., SHARMA D. M., VAIDYA A., VISHNU S. R. *et al.* (2017). The Hindi/Urdu Treebank Project. In *Handbook of Linguistic Annotation*. Springer Press.

CHAI Y., LIANG Y. & DUAN N. (2022). Cross-Lingual Ability of Multilingual Masked Language Models: A Study of Language Structure. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 4702–4712, Dublin, Ireland: Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.322.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8440–8451, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.747.

DE VRIES W., VAN CRANENBURGH A., BISAZZA A., CASELLI T., VAN NOORD G. & NISSIM M. (2019). BERTje: A Dutch BERT Model. arXiv:1912.09582 [cs], DOI : 10.48550/arXiv.1912.09582.

DESHPANDE A., TALUKDAR P. & NARASIMHAN K. (2022). When is BERT Multilingual? Isolating Crucial Ingredients for Cross-lingual Transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 3610–3623, Seattle, United States: Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.264.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota: Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DHAMECHA T., MURTHY R., BHARADWAJ S., SANKARANARAYANAN K. & BHATTACHARYYA P. (2021). Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 8584–8595, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.675.

FREIHAT A. A. & ABBAS M., Éds. (2019). *Proceedings of the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019 - Short Papers*. Trento, Italy: Association for Computational Linguistics.

GOLDHAHN D., ECKART T. & QUASTHOFF U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, p. 759–765, Istanbul, Turkey: European Language Resources Association (ELRA).

JOSHI R. (2022). L3Cube-HindBERT and DevBERT: Pre-Trained BERT Transformer models for Devanagari based Hindi and Marathi Languages. *arXiv preprint arXiv:2211.11418*.

K K., WANG Z., MAYHEW S. & ROTH D. (2020). Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In *International Conference on Learning Representations*. DOI : 10.48550/arXiv.1912.07840.

KAKWANI D., KUNCHUKUTTAN A., GOLLA S., N.C. G., BHATTACHARYYA A., KHAPRA M. M. & KUMAR P. (2020). IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, p. 4948–4961, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.445.

KALYAN K. S., RAJASEKHARAN A. & SANGEETHA S. (2021). AMMUS: A Survey of Transformer-based Pretrained Models in Natural Language Processing. arXiv:2108.05542 [cs], DOI : 10.48550/arXiv.2108.05542.

KHANUJA S., BANSAL D., MEHTANI S., KHOSLA S., DEY A., GOPALAN B., MARGAM D. K., AGGARWAL P., NAGIPOGU R. T., DAVE S. *et al.* (2021). Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

KHEMCHANDANI Y., MEHTANI S., PATIL V., AWASTHI A., TALUKDAR P. & SARAWAGI S. (2021). Exploiting Language Relatedness for Low Web-Resource Language Model Adaptation: An Indic Languages Study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 1312–1323, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.105.

KUMAR M A. (2019). NITK-IT_NLP@NSURL2019: Transfer Learning based POS Tagger for Under Resourced Bhojpuri and Magahi Language. In *Proceedings of the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019 - Short Papers*, p. 68–72, Trento, Italy: Association for Computational Linguistics.

LIN Y.-H., CHEN C.-Y., LEE J., LI Z., ZHANG Y., XIA M., RIJHWANI S., HE J., ZHANG Z., MA X., ANASTASOPOULOS A., LITTELL P. & NEUBIG G. (2019). Choosing Transfer Languages for Cross-Lingual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3125–3135, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1301.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE E., SEDDAH D. & SAGOT B. (2020). CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

MICALLEF K., GATT A., TANTI M., VAN DER PLAS L. & BORG C. (2022). Pre-training Data Quality and Quantity for a Low-Resource Language: New Corpus and BERT Models for Maltese. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, p. 90–101. arXiv:2205.10517 [cs], DOI : 10.18653/v1/2022.deeplo-1.10.

MULLER B., ANASTASOPOULOS A., SAGOT B. & SEDDAH D. (2021). When Being Unseen from mBERT is just the Beginning: Handling New Languages With Multilingual Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 448–462, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.38.

MUNDOTIYA R. K., SINGH M. K., KAPUR R., MISHRA S. & SINGH A. K. (2021). Linguistic Resources for Bhojpuri, Magahi, and Maithili: Statistics about Them, Their Similarity Estimates, and Baselines for Three Applications. *ACM Transactions on Asian and Low-Resource Language Information Processing*, **20**(6), 95:1–95:37. DOI : 10.1145/3458250.

OJHA A. K. (2019). English-Bhojpuri SMT System: Insights from the Karaka Model. arXiv:1905.02239 [cs], DOI : 10.48550/arXiv.1905.02239.

OJHA A. K., MALYKH V., KARAKANTA A. & LIU C.-H. (2020). Findings of the LoResMT 2020 Shared Task on Zero-Shot for Low-Resource languages. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, p. 33–37, Suzhou, China: Association for Computational Linguistics.

ORTIZ SUÁREZ P. J., ROMARY L. & SAGOT B. (2020). A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 1703–1714, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.156.

PALMER M., BHATT R., NARASIMHAN B., RAMBOW O., SHARMA D. M. & XIA F. (2009). Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, p. 14–17.

PATIL V., TALUKDAR P. & SARAWAGI S. (2022). Overlap-based Vocabulary Generation Improves Cross-lingual Transfer Among Related Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 219–233, Dublin, Ireland: Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.18.

PIRES T., SCHLINGER E. & GARRETTE D. (2019). How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4996–5001, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1493.

PRIYADARSHI A. & SAHA S. K. (2020). Towards the first Maithili part of speech tagger: Resource creation and system development. *Computer Speech & Language*, **62**, 101054. DOI : 10.1016/j.csl.2019.101054.

PROISL T., UHRIG P., BLOMBACH A., DYKES N., HEINRICH P., KABASHI B. & MAMMARELLA S. (2019). The_Illiterati: Part-of-Speech Tagging for Magahi and Bhojpuri without Even Knowing the Alphabet. In *Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019-Short Papers*, p. 73–79.

RI R. & TSURUOKA Y. (2022). Pretraining with Artificial Language: Studying Transferable Knowledge in Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 7302–7315, Dublin, Ireland: Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.504.

ULČAR M. & ROBNIK-ŠIKONJA M. (2020). FinEst BERT and CroSloEngual BERT. In P. SOJKA, I. KOPEČEK, K. PALA & A. HORÁK, Éds., *Text, Speech, and Dialogue*, p. 104–111, Cham: Springer International Publishing.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., LE SCAO T., GUGGER S., DRAME M., LHOEST Q. & RUSH A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, p. 38–45, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-demos.6.

WU S. & DREDZE M. (2019). Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 833–844, Hong Kong, China: Association for Computational Linguistics. DOI : 10.18653/v1/D19-1077.

ZAMPIERI M., NAKOV P., LJUBEŠIĆ N., TIEDEMANN J., MALMASI S. & ALI A., Éds. (2018). *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Pauzee : Prédiction des pauses dans la lecture d'un texte

Marion Baranes, Karl Hayek, Romain Hennequin et Elena V. Epure[1]
(1) Deezer Research, Paris, France
research@deezer.com

RÉSUMÉ

Les pauses silencieuses jouent un rôle crucial en synthèse vocale où elles permettent d'obtenir un rendu plus naturel. Dans ce travail, notre objectif consiste à prédire ces pauses silencieuses, à partir de textes, afin d'améliorer les systèmes de lecture automatique. Cette tâche n'ayant pas fait l'objet de nombreuses études pour le français, constituer des données d'apprentissage dédiées à la prédiction de pauses est nécessaire. Nous proposons une stratégie d'inférence de pauses, reposant sur des informations temporelles issues de données orales transcrites, afin d'obtenir un tel corpus. Nous montrons ensuite qu'à l'aide d'un modèle basé sur des transformeurs et des données adaptées, il est possible d'obtenir des résultats prometteurs pour la prédiction des pauses produites par un locuteur lors de la lecture d'un document.

ABSTRACT

Pauzee : Pauses Prediction in text reading.

Silent pauses play a crucial role in text-to-speech synthesis, where they help make the text reading sound more natural. In this work, our goal is to predict these silent pauses from texts to improve automatic reading systems. As this task has not been extensively studied for French, it is necessary to build training data dedicated to the prediction of pauses. We propose a strategy for inferring pauses, based on temporal information from transcribed speech, in order to obtain such a corpus. We then show that with the help of a model based on transformers and appropriate data, it is possible to obtain promising results for the prediction of pauses produced by a speaker during text reading.

MOTS-CLÉS : pauses silencieuses, prédiction des pauses, annotation pour la synthèse vocale.

KEYWORDS: silent break, break prediction, speech synthesis annotation.

1 Introduction

La conversion de textes en contenu audio permet de multiples applications telles que la production de nouveaux médias sonores (*e.g.* livres audio) (Steinhaeusser *et al.*, 2021) ou l'amélioration de l'accessibilité, pour les personnes malvoyantes ou non-voyantes, à des contenus textuels (Freitas & Kouroupetroglou, 2008). Toutefois, obtenir une lecture réaliste et fluide du texte reste un défi en synthèse vocale. Par exemple, sans variations d'emphase, de ton, de rythme ou encore sans pauses de respiration, un texte lu automatiquement semblera monotone et peu naturel (Székely *et al.*, 2020).

Dans ce projet, nous nous intéressons aux pauses silencieuses réalisées par les lecteurs en français, dans le but d'améliorer à terme la lecture automatique d'histoires et d'articles. Nous cherchons notamment à identifier les endroits où les humains marquent une pause silencieuse, consciemment ou non, lors de leur lecture d'un texte à voix haute. Ces pauses peuvent être produites pour des raisons

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 43–55.

respiratoires, stylistiques ou encore syntaxiques (Grosman *et al.*, 2018). Elles peuvent être de durée variable. Campione & Véronis (2002b) ainsi que Goldman *et al.* (2010) distinguent par exemple les pauses brèves, moyennes et longues. Grosman *et al.* (2018) définissent les pauses silencieuses comme "une interruption de la phonation". Dans ce papier, nous considérons qu'une interruption de la phonation est une pause silencieuse dès lors qu'elle peut être perçue à l'oreille humaine. Cette notion de perception est délicate à prendre en compte. Pour ce faire, certain travaux choisissent par exemple d'écarter les pauses trop brèves en utilisant des seuils prédéfinis en millisecondes (Grosjean & Deschamps, 1975; Candea, 2000). Toutefois, étant donnée la variabilité des débits de paroles et des durées de pauses, utiliser des seuils fixes peut avoir des conséquences sur les résultats (Campione & Véronis, 2002b; Grosman *et al.*, 2018). Utiliser la ponctuation comme marqueurs de pauses et de respiration est aussi une solution proposée (Wang *et al.*, 2021) pour prédire où un lecteur fera des pauses. Néanmoins, elle peut ne pas convenir à tous les types de textes, notamment les phrases longues et peu ponctuées comme dans les articles de Wikipédia. Définir quand une pause silencieuse peut apparaître est ainsi un problème non trivial.

Comme nous le verrons en section 2, les stratégies utilisées ont beaucoup évolué au fil du temps. Toutefois, ces études ont majoritairement été effectuées pour la langue anglaise (Székely *et al.*, 2019). D'autres travaux se sont penchés sur des tâches similaires, telles que la prédiction de ponctuation (Rei *et al.*, 2021) avec des approches multilingues. Chacun de ces systèmes de prédiction requiert des données d'apprentissage pertinentes dans la langue traitée. Ces données, très spécifiques, peuvent se révéler complexes à trouver pour des langues moins dotées que l'anglais, telles que le français. Or, les construire de toutes pièces n'est pas une tâche aisée. Comme expliqué plus haut, un système se reposant uniquement sur des symboles de ponctuation ou sur des seuils fixés manuellement ne semble pas optimal. Pour pallier cela, nous proposons d'inférer un tel corpus, à partir de données temporelles intermots provenant de transcriptions de données orales, et de l'utiliser pour prédire des pauses. Le système de prédiction de pauses proposé, *Pauzee*, est inspiré des systèmes de prédiction de ponctuation. L'adaptation de tels systèmes à notre tâche permettrait en effet de prédire, dans un texte, les endroits où des pauses pourraient être réalisées en lecture. Dans un second temps, nous nous intéressons au niveau de détail qu'il est possible d'obtenir en tentant de prédire la longueur d'une pause. En synthèse vocale, si toutes les pauses générées sont de la même durée, le résultat restera peu naturel voire peu intelligible. Disposer d'informations à ce sujet est donc primordial. Les contributions principales réalisées dans ce travail sont les suivantes [1]

— La mise en place d'un système d'inférence dynamique des pauses silencieuses à partir d'informations temporelles prenant en considération les variations de ces pauses compte-tenu des différents locuteurs et types de corpus, et permettant ainsi la production d'un corpus annoté.
— Pauzee : l'implémentation d'une nouvelle approche s'appuyant sur des transformeurs pour la prédiction de pauses réalisées lors de la lecture d'un texte.

La suite du papier est organisée comme suit : la section 2 offre un aperçu des travaux déjà faits dans le domaine. La section 3 détaille la construction des données utilisées pour l'apprentissage et l'évaluation du système de prédiction Pauzee. Ce système est ensuite décrit en section 4. Enfin, les résultats obtenus sont détaillés en section 5 et sont suivis d'une conclusion en section 6.

1. Le code développé pour la création du jeu de données utilisé ainsi que pour la prédiction de pauses silencieuses est disponible ici : https://github.com/deezer/pauzee_taln23.

2 État de l'art

Dans le domaine du traitement automatique des langues, la prédiction de pauses a donné lieu à plusieurs travaux. Les premiers sur le sujet s'appuyaient sur des systèmes par règles (Sorin *et al.*, 1987; Bachenko & Fitzpatrick, 1990; Atterer, 2002) et sur des arbres de décision (Ostendorf & Veilleux, 1994; Apel *et al.*, 2004). Puis, l'évolution des techniques en apprentissage automatique a influencé le domaine. Certaines études ont ainsi opté pour des modèles de Markov cachés (Taylor & Black, 1998), des champs aléatoires conditionnels (Keri *et al.*, 2007) ou encore des réseaux de neurones récurrents (Pascual & Bonafonte, 2016). Plus récemment, Székely *et al.* (2019) a choisi d'utiliser un système de classification qui s'appuie sur un réseau de neurones avec deux couches convolutives suivies d'une couche récurrente bidirectionnelle de type bloc LSTM. Cette stratégie permet de prendre en compte un plus long contexte temporel et d'obtenir de meilleures performances. Les auteurs ont notamment repris cette approche dans des travaux ultérieurs (Székely *et al.*, 2020; Alexanderson *et al.*, 2020; Wang *et al.*, 2021).

Toutes ces approches nécessitent des données d'apprentissage dans la langue étudiée. Ces données ne sont pas systématiquement disponibles dans toutes les langues. Pour répondre à ce problème, différentes stratégies sont mises en place. Certaines études prennent le parti d'utiliser des corpus existants, il en existe notamment pour l'anglais. Nous pouvons par exemple citer le Spoken English Corpus qui est annoté en pauses (Taylor & Black, 1998). D'autres, choisissent d'annoter manuellement leurs données (Ostendorf & Veilleux, 1994; Székely *et al.*, 2019). Enfin, une dernière stratégie est de réaliser cette annotation de manière plus automatique. C'est par exemple le cas de Keri *et al.* (2007) qui considèrent tous les silences de plus de 150 ms comme des pauses silencieuses. À notre connaissance, à l'exception de Sorin *et al.* (1987), rares sont les travaux fait sur le français et, par conséquent, rares sont les corpus annotés disponibles.

Comme dit précédemment, les symboles de ponctuation, tels que les points et les virgules, sont souvent associés à une pause dans la lecture (Wang *et al.*, 2021). Campione & Véronis (2002a) montrent d'ailleurs que près de 88% des pauses lors de la lecture apparaissent en présence d'une ponctuation (Campione & Véronis, 2002a). Cette même étude constate que 18,7% des symboles de ponctuations ne provoquent pas de pauses. Bien que non systématique, cette corrélation entre ponctuation et pauses reste observable et suggère que la prédiction de pauses pourrait utiliser des méthodes similaires à celles de la prédiction de ponctuation. En prédiction de ponctuation, les travaux les plus récents utilisent majoritairement des transformeurs. C'est par exemple le cas de Sunkara *et al.* (2020) qui prédit la ponctuation en alignant des caractéristiques lexicales et prosodiques. Plus récemment, SEPP-NLG 2021 (Tuggener & Aghaebrahimian, 2021), la première tâche partagée sur la prédiction de fin de phrase et de ponctuation, a été organisée afin de développer des solutions dans ce domaine. Trois systèmes se sont retrouvés gagnants : OnPoint (Michail *et al.*, 2021), HTW+t2k (Guhr *et al.*, 2021) et Unbabel (Rei *et al.*, 2021). Ces trois systèmes avaient pour point commun d'utiliser des transformeurs.

Pauzee, le système de prédiction de pauses silencieuses que nous voulons mettre en place à besoin de données d'apprentissage. Nous proposons dans ce travail de développer une nouvelle manière d'inférer si un silence annoté en millisecondes peut être considéré comme une pause silencieuse. Pour ce faire nous faisons cet apprentissage sur des données orales transcrites qui ont l'avantage d'être disponibles. Contrairement à Keri *et al.* (2007), nous ne souhaitons pas utiliser un seuil fixe prédéfini. Un telle stratégie risquerait d'ignorer les variations de durée des pauses observables d'un locuteur à un autre (Grosman *et al.*, 2018). Pour éviter cela, nous proposons d'apprendre ces seuils

de manière dynamique, les rendant ainsi adaptables au style de narration du locuteur. À la vue des travaux réalisés dans le domaine, nous avons choisi de nous inspirer des travaux les plus récents et d'évaluer l'intérêt des transformeurs pour une telle tâche. Des travaux en prédiction de ponctuation ayant déjà été réalisés avec des transformeurs, nous proposons de reprendre cette stratégie pour l'adapter à la prédiction de pauses silencieuses.

3 Données

Afin de constituer nos données d'entraînement et d'évaluation, nous proposons d'inférer les pauses produites par les locuteurs à partir de deux corpus de transcriptions d'histoires parlées et lues en français, contenant des données temporelles indiquant le début et la fin de chaque mot. Le premier, le corpus SynPaFlex (Sini *et al.*, 2018), contient des extraits de livres datant du 19ème siècle lus par une seule et unique locutrice. De ce corpus, seuls les textes contenant des informations temporelles ont été conservés [2]. Le second corpus, contenu sur la plate-forme ORFEO (Outils et Ressources sur le Français Ecrit et Oral) (Benzitoun & Debaisieux, 2020), est le French Oral Narrative Corpus (Carruthers *et al.*, 2013). Ce jeu de données regroupe 87 contes oraux narrés en français par des conteurs professionnels et semi-professionnels. Chaque conte n'apparaît qu'une fois dans le corpus, nous ne disposons pas de contes identiques racontés par deux locuteurs différents. Notons que ce corpus est un corpus de parole spontanée ce qui le différencie de SynPaFlex. Bien que le format conté de mémoire des contes se distingue d'un point de vue prosodique du format lu (Levin *et al.*, 1982), il se rapproche toutefois de ce dernier d'un point de vue intentionnel. En effet dans ces deux cas, il est important de pouvoir partager une histoire de manière compréhensible et les pauses y jouent un rôle clé, bien que leur durée et leur fréquence différent. Il est à préciser qu'un prétraitement a été réalisé sur ces deux corpus : tous les symboles de ponctuation ont été retirés. Ce pré-traitement avait pour but d'aligner les deux corpus ensemble, le corpus conté n'étant pas doté de ponctuation. Par ailleurs, retirer la ponctuation de nos corpus présente aussi un intérêt pour notre étude puisque nous ne souhaitons pas développer un système dépendant des signes de ponctuation.

Pour annoter ces corpus, nous proposons d'utiliser les informations temporelles disponibles pour inférer la présence des pauses. Ces informations nous indiquent notamment la durée (en millisecondes) de chaque silence produit entre deux mots. Nous appellerons ici cette information "silence intermot". Précisons que tous les silences intermots, trop courts, ne peuvent être considérés comme des pauses (*e.g.* les occlusives). Toutefois, fixer un seuil strict pour la prédiction de pauses (tel que 150, 200 ou 300ms) n'est pas recommandé en raison de la variabilité temporelle des pauses et de la façon de s'exprimer du locuteur (Campione & Véronis, 2002a; Grosman *et al.*, 2018). Sans seuil, Campione & Véronis (2002a) observent d'ailleurs manuellement des pauses descendant jusqu'à 60ms. Pour définir ce que nous pouvons considérer comme pause ou non, plusieurs analyses de corpus ont été réalisées. Bien que SynPaFlex se distingue du corpus French Oral Narrative Corpus (pauses plus courtes et moins nombreuses), nous ne pouvons savoir si ces différences sont dues au type de corpus ou à la manière de s'exprimer de la locutrice. Pour ces raisons, les observations sur les pauses, décrites ci-dessous, se concentrent sur le corpus Oral Narrative.

La figure 1 présente l'évolution de la durée des silences intermots observés dans le French Oral Narrative, percentile par percentile, pour chaque locuteur. Les courbes représentent les différents

2. Les textes conservés sont : *la fille du pirate*(1878) de Chevalier ; *la vampire* (1865) de Feval, *Madame Bovary* (1857) de Flaubert, *Carmen* (1845) et la vénus d'Ille (1835) de Mérimée, *les mystères de Paris* (1842) de Sue et, *Contes Sénégal et du Niger* (1913) de Zeltner.

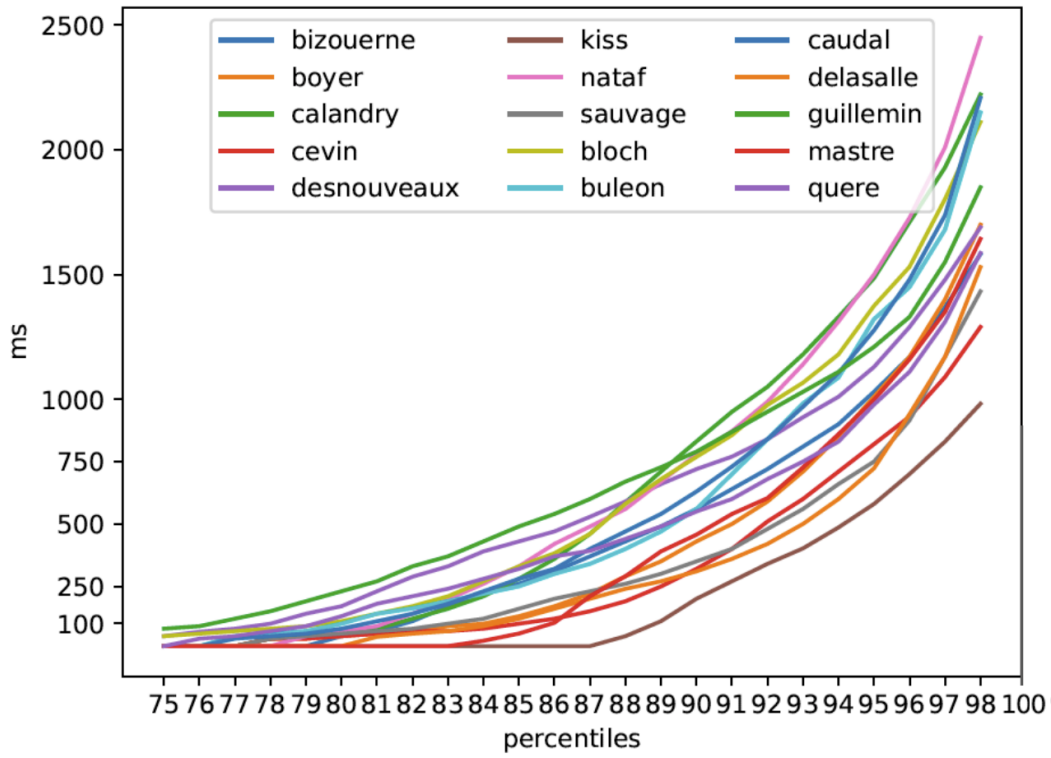


FIGURE 1 – Répartition des pauses faites dans le corpus Oral Narrative

locuteurs, l'ordonnée correspondant au nombre de millisecondes et l'abscisse indiquant les percentiles. Cette visualisation est limitée aux percentiles supérieurs à 75, les durées intermots en dessous étant trop courtes pour être considérées comme des pauses silencieuses. En effet, à l'exception d'un locuteur, à 75 percentiles tous les silences intermots produits sont en dessous de 60ms. Cette illustration offre deux constats : 1) les locuteurs ne font pas le même pourcentage de pauses dans leurs silences intermots (ce pourcentage varie entre 22% et 12%) et 2) la durée d'une pause peut varier d'une personne à l'autre. Par exemple, les pauses les plus longues constatées pour chaque locuteur varient entre 1000ms et 2500ms. Ces observations doivent être interprétées avec prudence : les différences peuvent également provenir du contenu de l'histoire racontée par chaque locuteur, chaque récit étant différent des autres. Suite à cette étude, nous avons choisi de définir un seuil dynamique pour chaque locuteur et chaque histoire. Utiliser un seuil ne dépendant pas d'un nombre de millisecondes prédéfini pourrait offrir une meilleur prise en compte de la fluctuation des pauses et ainsi conduire à une lecture plus fluide et naturelle. Pour cela, nous appuyons sur deux critères : la taille de la pause en millisecondes et le percentile. Suite à notre étude et à la lecture de différentes études citées précédemment, nous considérons qu'une pause doit durer au minimum 80 ms et qu'un narrateur peut faire jusqu'à 22% de pauses silencieuses parmi les silences intermots observés. Pour chaque corpus, nous avons observé la longueur de toutes les pauses apparaissant à partir du percentile 78, puis avons incrémenté ce percentile jusqu'à trouver des longueurs de pauses correspondant à ces critères. Cette stratégie nous permet d'avoir un corpus annoté en pauses prenant en considération leur variabilité compte-tenu du locuteur et du corpus.

Parmi les pauses réalisées en texte lu, se distinguent les pauses brèves, moyennes et longues (Campione & Véronis, 2002b; Goldman *et al.*, 2010; Grosman *et al.*, 2018). Cependant, les seuils qui délimitent

ces pauses varient selon les études : celle de la pause courte est souvent comprise entre 100 ms et 250 ms (Campione & Véronis, 2002b; Goldman *et al.*, 2010; Bailly & Gouvernayre, 2012), celle de la pause moyenne entre 500 ms et 1000 ms (Goldman *et al.*, 2010; Campione & Véronis, 2002b) et à la pause longue correspond aux pauses de durées supérieures (Campione & Véronis, 2002b; Goldman *et al.*, 2010). De même que le seuil qui sépare un silence d'une pause peut varier, nous pensons que les seuils qui distinguent une pause courte, moyenne ou longue peuvent varier selon les locuteurs. On constate sur la figure, il n'y a pas de séparation nette entre les différentes longueurs de pauses, un constat qu'avait également fait Campione & Véronis (2002b). Pour chaque locuteur, nous déterminons les percentiles parmi lesquels se trouvent les pauses, puis nous divisons ces percentiles en trois groupes de tailles égales afin d'obtenir les seuils correspondant aux pauses courtes, moyennes et longues. En moyenne, les seuils séparant les pauses courtes des moyennes sont de 350ms et celles séparant les moyennes des longues est de 950ms. Cette seconde stratégie permet de compléter le corpus annoté en pause par une seconde annotation en longueur de pauses.

La table 1 illustre le nombre de pauses courtes (C), moyennes (M) et longues (L) ainsi que leur total dans chaque corpus. À noter, les corpus ont été divisés au préalable en deux afin de constituer un corpus dédié à l'entraînement (train) et un autre dédié à l'évaluation (test). Pour SynPaFlex, cette division a été effectuée en extrayant 30% des lignes de chaque corpus pour constituer le corpus test et en conservant les autres pour le corpus train. Pour le French Oral Narrative Corpus, une division par ligne n'était pas possible, chaque fichier contenant un mot par ligne. Ainsi, nous avons procédé à la séparation des fichiers de contes en veillant, tant que possible, à ce que les mêmes locuteurs apparaissent à la fois dans les corpus de test et d'entraînement. Pour ce dernier cas, nous avons tenté d'avoir ici aussi un corpus train représentant 70% du corpus initial et un corpus test représentant 30% du corpus initial.

TABLE 1 – Nombre de pauses et de mots contenus dans chaque corpus.

Datasets	# Pauses C	# Pauses M	# Pauses L	# Pauses	# Mots
SynPaflex train	2 702	3 164	3 047	8 913	54 426
Oral Narrative train	6 272	7 481	7 072	20 825	115 912
Total train	8 974	10 645	10 119	29 738	170 338
SynPaflex test	874	1 062	1 017	2 953	19 453
Oral Narrative test	2 659	3 101	2 960	8 720	48 747
Total test	3 533	4 163	3 977	11 673	68 200

4 Système d'apprentissage

Pour cette tâche, nous souhaitons reprendre un système faisant de la prédiction de ponctuation pour l'adapter à notre tâche : la prédiction de pauses silencieuses et de leur longueur. La sélection du système qui nous a servi de point de départ s'est faite sur deux critères : ce système devrait être au niveau de l'état de l'art pour la prédiction de ponctuation et devait être très performant pour le français. Notre choix s'est arrêté sur Unbabel (Rei *et al.*, 2021) qui utilise un modèle multilingue (anglais, français, allemand et italien) proposant de prédire dans un texte les fins de phrases et les signes de ponctuation. Son système a été retenu gagnant à SEPP-NLG 2021 (Tuggener & Aghaebrahimian, 2021) et a obtenu de bons résultats pour le français.

Le système d'Unbabel étend le travail de Miguel Guerreiro *et al.* (2021). Il fonctionne comme suit : un encodeur pré-entraîné reposant sur un transformeur est tout d'abord mis en place. Cela permet la création d'embeddings pour chaque sous-mot et chaque couche du transformeur. Puis ils encapsulent toutes les couches de ce transformeur en un seul embedding à l'aide d'un mécanisme d'attention. Enfin, la dernière étape est dédiée aux têtes de classification qui concatènent les embeddings obtenus. Cela permet de les utiliser comme paramètres afin de prédire d'une part la fin d'une phrase et d'autre part le signe de ponctuation. Pour entraîner leur modèle, Rei *et al.* (2021) utilisent un système d'affinage. Ils combinent ainsi le modèle XLM-Roberta large (Conneau *et al.*, 2020) à un jeu de données annoté pour la prédiction de fin de phase et ponctuation (corpus fourni par les organisateurs de SEPP-NLG 2021 (Tuggener & Aghaebrahimian, 2021)). Les paramètres, bien décrits dans le papier, sont divisés en deux parties : ceux pour XLM-Roberta large et ceux pour les têtes classification. Les paramètres de l'encodeur sont gelés durant les étapes de 0,1% de la première époque. Cela permet aux paramètres pour la classification de s'ajuster à l'objectif de la tâche avant de modifier, et ainsi affiner, ceux pré-entraînés. Entre chaque époque une évaluation est effectuée sur 50% des données. Si aucune amélioration n'est constatée pendant deux époques consécutives, l'entraînement est interrompu.

Notre modèle, Pauzee, reprend ce système et l'adapte à la prédiction de pause. Pour ce faire, nous avons principalement modifié les paramètres qui concernaient la tâche de classification. Nous avons ainsi remplacé les paramètres propres à la ponctuation par des paramètres propres aux pauses. Nous prenons ainsi en compte la prédiction de pauses avec deux valeurs possibles : "absence de pause" et "présence de pause" et la prédiction de la longueur d'une pause avec quatre valeurs possibles : "absence de pause", "pause courte", "pause moyenne" et "pause longue". Concernant l'entraînement nous continuons à utiliser le modèle de langue XLM-Roberta large que nous affinons, non plus au jeu de données fourni par SEPP-NLG 2021, mais aux deux jeux de données (French Oral Narrative Corpus et SynPaFlex) automatiquement annotés en pauses décrits dans la section 3, plus adaptés à notre tâche.

5 Résultats

Ce travail tente de comprendre dans quelle mesure des pauses silencieuses, produites par un humain à partir d'un texte lu, ainsi que leurs longueurs peuvent être prédites. L'évaluation de ces tâches de classification est réalisée avec des métriques classiques : la précision, le rappel et la F-mesure. Ces métriques sont calculées pour chacune des classes étudiées, leur macro-moyenne et leur moyenne pondérée.

Prédire la présence d'une pause à un endroit donné est une tâche de classification binaire. Nous ne disposons malheureusement pas des modèles et des corpus utilisés par les autres travaux décrits dans l'état de l'art et nous ne pouvons donc comparer notre système, Pauzee, aux leurs. Par conséquent, nous proposons d'utiliser deux approches de base :

— La première, "Syllabe", est naïve. Elle repose sur l'idée que les locuteurs font des pauses régulières. Grosjean & Deschamps (1973) montrent que la longueur médiane des espaces entre les pauses sont de 6 syllabes dans une description et de 15 syllabes dans une interview. Nous proposons donc d'ajouter une pause de manière régulière toutes les 7 syllabes. Pour déterminer le nombre de syllabes pris en compte, nous avons testé cette approche de base en allant de 5 à 15 syllabes. Plus le nombre de syllabes était élevé, meilleurs étaient les résultats pour la prédiction d'absence de pauses et moins bons étaient ceux pour la prédiction de pauses.

Le choix concernant nombre de syllabes s'est donc fait sur la macro-moyenne.

— La seconde, "Unbabel", propose de s'appuyer sur le système Unbabel initial entraîné pour de la prédiction de ponctuation. L'idée sous-jacente étant de mesurer l'apport des données d'entraînement que nous avons choisi d'utiliser pour affiner le modèle appris. Unbabel prédit non pas des pauses, mais de la ponctuation. Ainsi, pour prendre en compte ses résultats dans notre évaluation, tous les marqueurs de fin de phrases (".", " !", " ?") ainsi que les virgules et point virgule prédits par Unbabel sont considérés comme une prédiction de pause.

Le tableau 2 montre, pour chaque classe prédite et en moyenne, les résultats obtenus par les approches de base et par Pauzee. La dernière colonne indique le nombre d'éléments concernés.

TABLE 2 – Résultats pour la prédiction de pauses.

Système		Précision	Rappel	F-Mesure	# éléments
Syllabe	classe Absence Pause	0,84	0,81	0,83	56 489
	classe Pause	0,23	0,27	0,25	11 673
	macro-moyenne	0,53	0,54	0,54	68 162
	moyenne pondérée	0,74	0,72	0,73	68 162
Unbabel	classe Absence Pause	0,92	0,93	0,93	56 489
	classe Pause	0,65	0,63	0,64	11 673
	macro-moyenne	0,79	0,78	0,79	68 162
	moyenne pondérée	0,88	0,88	0,88	68 162
Pauzee	classe Absence Pause	0,93	0,95	0,94	56 489
	classe Pause	0,73	0,64	0,68	11 673
	macro-moyenne	0,83	0,79	0,81	68 162
	moyenne pondérée	0,89	0,90	0,90	68 162

Plusieurs études ont montré que les pauses apparaissaient aux niveaux des frontières syntaxiques (Grosman *et al.*, 2018). Dès lors, les résultats obtenus pas la "Syllabe" ne sont pas surprenant et ce d'autant plus sans prise en compte des débuts et fins de phrases (indication absente de nos corpus). En effet, les transformeurs utilisés par les systèmes Unbabel et Pauzee apprennent des représentations vectorielles de mots et de phrases à partir de grandes quantités de textes. Cela leur permet de prendre en compte de manière implicite de nombreuse informations syntaxiques telles que ces frontières. Concernant la tâche de prédiction de l'absence de pause, Unbabel et Pauzee obtiennent des résultats similaires. Concernant la tâche de prédiction d'une pause, les résultats obtenus sont plus éloquents au niveau de la précision obtenue par les deux tâches. Celui obtenu par Pauzee monte à 0,73 alors que celui d'Unbabel reste à 0,65. Le rappel obtenu montre que plusieurs pauses ne sont toujours pas prédites et que ce système d'apprentissage pourrait être encore amélioré. Cela s'explique notamment par le fait que certaines pauses sont plus propres au locuteur qu'au texte lu. Notons toutefois que le fait que le système Unbabel, appris sur de la ponctuation, et le notre, appris sur des pauses silencieuses, obtiennent des résultats similaires illustre bien la corrélation existante entre pause et ponctuation.

Quelques exemples de résultats des systèmes Unbabel et Pauzee sont partagés dans la table 3. Les exemples proviennent des corpus SynPaFlex ("*Carmen*" (1) et "*Madame Bovary*" (2)) et French Oral Narrative ("*Mélusine*" (3) et "*La pierre barbue*" (4)). Chacun de ces passages montrent les différentes découpes du texte en pauses <P> obtenues par Unbabel et par Pauzee. Pour certains passages, les deux propositions semblent acceptables et dépendent plutôt de la manière de raconter du locuteur. C'est par exemple, le cas des extraits 1 et 3. L'exemple 4 est intéressant. Il propose deux manières de raconter un même passage. Unbabel propose une version très ponctuée et contée. Notre système propose

TABLE 3 – Exemples de pauses prédites .

Unbabel	1) *je me sentais près de pleurer* <P> *je lui dis que je reviendrais et je me sauvai* <P> 2) *le soir* <P> *après le maigre diner de son propriétaire* <P> *il remontait à sa chambre et se remettait au travail dans ses habits mouillés qui fumaient sur son corps* <P> *devant le poêle rougi* <P> 3) *on était samedi soir* <P> *Raymondin a reçu la nouvelle tout seul et il a eu toute la nuit pour y penser* <P> *et le dimanche matin* <P> *Mélusine est arrivée tout doucement vers lui* 4) *la hyène* <P> *elle* <P> *rôdait partout* <P> *affamée comme une bête* <P> *quand* <P> *un jour* <P> *elle est allée au fond d' une vallée* <P>
Pauzee	1) *je me sentais près de pleurer* <P> *je lui dis que je reviendrais* <P> *et je me sauvai* <P> 2) *le soir après le maigre diner de son propriétaire* <P> *il remontait à sa chambre* <P> *et se remettait* <P> *au travail dans ses habits mouillés* <P> *qui fumaient sur son corps* <P> *devant le poêle rougi* <P> 3) *on était samedi soir* <P> *Raymondin a reçu la nouvelle tout seul* <P> *et il a eu toute la nuit pour y penser* <P> *et le dimanche matin* <P> *Mélusine est arrivée tout doucement vers lui <P>* 4) *la hyène elle rôdait partout affamée comme une bête* <P> *quand un jour elle est allée au fond d' une vallée <P>*

une version plus épurée, peut être trop pour paraître naturel. L'exemple 2 est plus problématique : Unbabel ne le segmente pas assez et nous fait prononcer 19 mots d'affilée sans aucune pause. Pauzee, lui, le segmente trop. Il propose par exemple une pause après le mot "remettait", ce qui gène la compréhension. Pour être acceptable cette pause devrait être à peine perceptible. Ces exemples nous montrent ainsi les limites des deux systèmes.

La seconde question à laquelle nous souhaitons répondre est relative à la longueur des pauses. Elle tend à préciser notre compréhension des résultats présentés plus haut. Il s'agit, cette fois-ci, d'une tâche de classification en classes multiples. Le tableau 4 illustre les résultats obtenus pour cette seconde tâche. Aux vues des résultats obtenus ici, on constate que la prédiction de l'absence de pauses concorde avec les résultats précédents mais que la prédiction des différents types de pauses est plus complexe à interpréter. La matrice de confusion proposée en figure 2 illustre de manière plus précises les résultats obtenus pour chaque classe. Cette matrice a été normalisée par ligne donc en fonction des pauses attendues.

TABLE 4 – Résultats pour la prédiction de longueur des pauses.

Système	Précision	Rappel	F-Mesure	# éléments
classe Absence Pause	0,90	0,98	0,94	56 489
classe Pause C	0,25	0,00	0,00	3 533
classe Pause M	0,33	0,15	0,21	4 163
classe Pause L	0,48	0,63	0,55	3 977
macro-moyenne	0,49	0,44	0,42	68 162
moyenne pondérée	0,81	0,86	0,82	68 162

On y constate tout d'abord que l'absence de pauses demeure bien prédite. C'est lors de la prédiction des pauses que Pauzee rencontre plus de difficultés. Les pauses courtes ne sont ici jamais prédites. Les pauses moyennes ne sont correctement prédites que pour 15% d'entre elles. Toutefois, on note que 48% d'entre elles sont bien reconnues en tant que pauses. Enfin, les pauses longues sont bien prédites pour 63% d'entre elles et 73% d'entre elles sont reconnues comme pauses. Ainsi, plus une pause est longue, plus elle semble évidente à prédire. Cela s'explique principalement par la variabilité des pauses. Les pauses longues sont probablement celles qui sont les plus communes à tous les

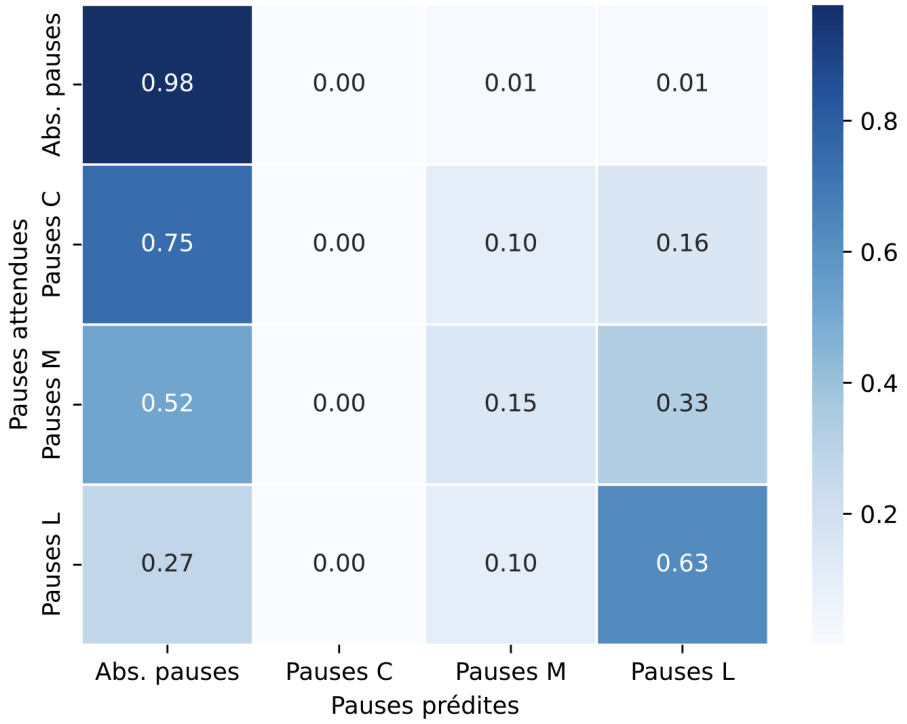


FIGURE 2 – Matrice de confusion des résultats normalisés

locuteurs. Il s'agit de pauses marquant souvent des fins de phrases ou de paragraphes. Leurs contextes d'apparition sont, par conséquent, plus réguliers. Les pauses courtes, quant à elles, semblent plus propres au locuteur. Elles apparaissent généralement au milieu des phrases et ont tendance à être optionnelles, ce qui rend leur prédiction moins claire. Par ailleurs, il est important de noter que plus une pause est brève moins elle est audible. On peut donc se demander si chaque pause courte de notre corpus serait annotée comme telle lors d'une tâche d'annotation manuelle. Les résultats obtenus dans la table 4 sont ainsi dus à cette variabilité mais aussi au fait que notre système d'évaluation est trop strict. Il considère de même importance une erreur portant sur la prédiction d'une pause et une portant sur une confusion entre deux longueurs de pauses. Avec un système plus flexible, capable de considérer que ne pas prédire une pause doit être plus sanctionné qu'une erreur portant sur sa longueur, la F-Mesure de la macro-moyenne peuvent monter à 0,59 et celle de la moyenne pondérée à 0,86.

6 Conclusion

Dans cet article, nous présentons un système de prédiction des pauses silencieuses qui peuvent survenir lors de la lecture à voix haute d'un texte. Cette prédiction peut être utilisée pour améliorer les outils de synthèse vocale. Notre système, Pauzee, utilise un modèle de prédiction de ponctuation basé sur des transformeurs. Une méthode qui, à notre connaissance, n'avait pas été proposée auparavant dans la littérature. Nous avons entraîné Pauzee sur des données inférées plutôt que manuellement annotées car il n'existe pas de données spécifiques pour la prédiction de pauses en français. Ce système d'inférence offre une meilleure prise en compte de la fluctuation des temps de pauses qui peuvent apparaître d'un

locuteur à l'autre et entre différents type de corpus. Cela nous permet d'obtenir des données annotées pour chaque locuteur. Peu bruyant, Pauzee produit des résultats encourageants. Nous pouvons noter les pauses prédites sont en bonne partie corrélées à de la ponctuation, les autres restent toutefois un défi. Nos résultats mettent par ailleurs en évidence la variabilité des pauses silencieuses. Étant donné que chaque élément de notre corpus est lu ou narré par une seule personne, il est difficile de distinguer les pauses spécifiques au locuteur de celles communes à tous les locuteurs. On constate cependant que plus une pause est longue plus elle est prévisible et probablement attendue de tous.

Nos résultats pourraient être améliorés à plusieurs niveaux. Apprendre à prédire plus finement les pauses communes à tous les locuteurs et celles qui sont plus optionnelles est une première piste. Comme nous avons pu le constater, parfois plusieurs annotations semblent acceptables à l'oreille humaine. Par conséquent, une évaluation humaine de ce travail serait pertinente. Même si nos résultats ne sont pas identiques à ceux attendus dans le corpus d'évaluation, ils pourraient finalement être perçus comme valides par des humains. En outre, les seuils dynamiques de notre système d'inférence pourrait être mieux adaptés. La définition de pauses courtes demeure ici encore trop ambiguë et trop permissive. Enfin, les caractéristiques du locuteur ne sont pas prises en compte dans ce travail, les ajouter dans notre système pourrait le rendre plus précis et réduire le nombre d'erreurs.

Références

ALEXANDERSON S., SZÉKELY É., HENTER G. E., KUCHERENKO T. & BESKOW J. (2020). Generating coherent spontaneous speech and gesture from text. In *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, p. 1–3.

APEL J., NEUBARTH F., PIRKER H. & TROST H. (2004). Have a break ! modelling pauses in german speech. In *KONVENS*, p. 5–12.

ATTERER M. (2002). Assigning prosodic structure for speech synthesis : a rule-based approach. In *Speech Prosody 2002, International Conference*.

BACHENKO J. & FITZPATRICK E. (1990). A computational grammar of discourse-neutral prosodic phrasing in english. *Computational Linguistics*, **16**, 155–170.

BAILLY G. & GOUVERNAYRE C. (2012). Pauses and respiratory markers of the structure of book reading. In *Interspeech 2012-13th Annual Conference of the International Speech Communication Association*, p. Thu–O9d.

BENZITOUN C. & DEBAISIEUX J.-M. (2020). Orféo : un corpus et une plateforme pour l'étude du français contemporain. HAL : hal-03011344.

CAMPIONE E. & VÉRONIS J. (2002a). Etude des relations entre pauses et ponctuations pour la synthèse de la parole à partir de texte. In *Actes de la 9ème conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, p. 177–186, Nancy, France : ATALA.

CAMPIONE E. & VÉRONIS J. (2002b). A large-scale multilingual study of silent pause duration. In *Speech prosody 2002, international conference*.

CANDEA M. (2000). *Contribution à l'étude des pauses silencieuses et des phénomènes dits" d'hésitation" en français oral spontané. Etude sur un corpus de récits en classe de français.* Thèse de doctorat, Université de la Sorbonne nouvelle-Paris III.

CARRUTHERS J. *et al.* (2013). French oral narrative corpus. *Oxford Text Archive Core Collection*.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8440–8451, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.747.

FREITAS D. & KOUROUPETROGLOU G. (2008). Speech technologies for blind and low vision persons. *Technology and Disability*, **20**, 135–156. DOI : 10.3233/TAD-2008-20208.

GOLDMAN J.-P., FRANÇOIS T., ROEKHAUT S., SIMON A. C. *et al.* (2010). Étude statistique de la durée pausale dans différents styles de parole. *Journées d'Etude sur la Parole (JEP)*.

GROSJEAN F. & DESCHAMPS A. (1973). Analyse des variables temporelles du français spontané. *Phonetica*, **28**(3-4), 191–226.

GROSJEAN F. & DESCHAMPS A. (1975). Analyse contrastive des variables temporelles de l'anglais et du français : vitesse de parole et variables composantes, phénomènes d'hésitation. *Phonetica*, **31**(3-4), 144–184.

GROSMAN I., SIMON A. C. & DEGAND L. (2018). Variation de la durée des pauses silencieuses : impact de la syntaxe, du style de parole et des disfluences. *Langages*, **211**, 13–40. DOI : 10.3917/lang.211.0013.

GUHR O., SCHUMANN A.-K., BAHRMANN F. & BÖHME H.-J. (2021). Fullstop : Multilingual deep models for punctuation prediction. In *Swiss Text Analytics Conference*.

KERI V., PAMMI S. C. & PRAHALLAD K. (2007). Pause prediction from lexical and syntax information. In *Proceedings of International Conference on Natural Language Processing (ICON)*.

LEVIN H., SCHAFFER C. A. & SNOW C. (1982). The prosodic and paralinguistic features of reading and telling stories. *Language and speech*, **25**(1), 43–54.

MICHAIL A., WEHRLI S. & BUCKOVÁ T. (2021). Uzh onpoint at swisstext-2021 : Sentence end and punctuation prediction in nlg text through ensembling of different transformers (short paper). In *Swiss Text Analytics Conference*.

MIGUEL GUERREIRO N., REI R. & BATISTA F. (2021). Towards better subtitles : A multilingual approach for punctuation restoration of speech transcripts. *Expert Systems with Applications*, **186**, 115740. DOI : 10.1016/j.eswa.2021.115740.

OSTENDORF M. & VEILLEUX N. M. (1994). A hierarchical stochastic model for automatic prediction of prosodic boundary location. *Computational Linguistics*, **20**(1), 27–54.

PASCUAL S. & BONAFONTE A. (2016). Prosodic break prediction with rnns. In *Advances in Speech and Language Technologies for Iberian Languages : Third International Conference, IberSPEECH*, p. 64–72. DOI : 10.1007/978-3-319-49169-1_7.

REI R., BATISTA F., GUERREIRO N. M. & COHEUR L. (2021). Multilingual simultaneous sentence end and punctuation prediction (short paper). In *Swiss Text Analytics Conference*.

SINI A., LOLIVE D., VIDAL G., TAHON M. & DELAIS-ROUSSARIE É. (2018). SynPaFlex-corpus : An expressive French audiobooks corpus dedicated to expressive speech synthesis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, p. 328–333, Miyazaki, Japan : European Language Resources Association (ELRA). HAL : hal-01826690.

SORIN C., LARREUR D. & LLORCA R. (1987). A rhythm-based prosodic parser for text-to-speech systems in french. *XIème Congrès International des Sciences Phonétiques*, p. 125–128.

STEINHAEUSSER S. C., SCHAPER P. & LUGRIN B. (2021). Comparing a robotic storyteller versus audio book with integration of sound effects and background music. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '21 Companion, p. 328–333, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3434074.3447186.

SUNKARA M., RONANKI S., BEKAL D., BODAPATI S. B. & KIRCHHOFF K. (2020). Multimodal semi-supervised learning framework for punctuation prediction in conversational speech. In *Interspeech 2020*, p. 4911–4915. DOI : 10.21437/Interspeech.2020-3074.

SZÉKELY É., HENTER G. E., BESKOW J. & GUSTAFSON J. (2020). Breathing and speech planning in spontaneous speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 7649–7653 : IEEE.

SZÉKELY É., HENTER G. E. & GUSTAFSON J. (2019). Casting to corpus : Segmenting and selecting spontaneous dialogue for tts with a cnn-lstm speaker-dependent breath detector. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 6925–6929 : IEEE.

TAYLOR P. & BLACK A. W. (1998). Assigning phrase breaks from part-of-speech sequences. *Comput. Speech Lang.*, **12**(2), 99–117. DOI : 10.1006/csla.1998.0041.

TUGGENER D. & AGHAEBRAHIMIAN A. (2021). The sentence end and punctuation prediction in nlg text (sepp-nlg) shared task 2021. In *Swiss Text Analytics Conference*.

WANG S., ALEXANDERSON S., GUSTAFSON J., BESKOW J., HENTER G. E. & SZ'EKELY E. (2021). Integrated speech and gesture synthesis. *Proceedings of the 2021 International Conference on Multimodal Interaction*.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Reconnaissance de défigements dans des tweets en français par similarité d'alignements textuels

RÉSUMÉ

Cet article propose une première approche permettant la reconnaissance automatique de défigements linguistiques dans un corpus de tweets. Les recherches portant sur le domaine du figement ont gagné en popularité depuis quelques décennies. De nombreux travaux dérivés de cette notion sont également apparus, portant sur le phénomène corollaire du défigement. Alors que les linguistes essayent de décrypter les modes de construction de ces exemples de créativité lexicale, peu de travaux de recherche en TAL s'y sont intéressés. La problématique qu'offre le cas du défigement est pourtant intéressante : des outils informatiques peuvent-ils être en mesure de reconnaître automatiquement un défigement ? Nous présentons ici une méthodologie basée sur des alignements de séquences réalisés sur diverses couches d'informations linguistiques. Cette méthodologie permet l'isolement de potentiels défigements au sein d'un corpus de tweets. Nous expérimentons ensuite une méthode de tri par similarité des défigements potentiels isolés.

ABSTRACT

Recognition of unfrozen expressions in french tweets using similarity measures

This paper proposes a first approach for the automatic recognition of unfrozen expressions in a corpus of tweets. Research on frozen expressions has been gaining in popularity for a few decades. Similarly, many works derived from this notion have emerged, dealing with the phenomenon of unfreezing. While linguists try to understand the modes of appearance of this phenomenon and its relation to the freezing effect, no research work in computer science has focused on it. However, the scientific question that arises with freezing/unfreezing is interesting : can computer tools automatically recognize an unfrozen expression ? We present here a methodology based on sequence alignments performed on various linguistic layers. This methodology allows the isolation of possible unfrozen expressions within a corpus of tweets. We then use different similarity methods to sort these possible unfrozen expressions.

MOTS-CLÉS : Figement linguistique, Expression Figées, Défigement, Alignement, Similarité.

KEYWORDS: Frozen expressions, Unfrozen expressions, Alignment, Similarity.

1 Introduction

Le figement est le phénomène par lequel une séquence de mots va se fixer, faisant ainsi perdre aux différents termes leur sens individuel au profit d'un sens global faiblement compositionnel. Cette notion figure au cœur de nombreux travaux, bien qu'il n'y ait pas de définition univoque de ce phénomène (Lamiroy, 2008). Il en résulte un manque de propriétés formelles fiables permettant la reconnaissance automatique de ces phénomènes. Toutefois, certains chercheurs ont développé des ressources lexicales et des méthodes consacrées à la reconnaissance automatique de séquences figées (ci-après SF) en français (Leclère, 2000; Mejri, 2005; Fort *et al.*, 2018, 2020) comme pour d'autres langues (Baptista *et al.*, 2004; Català & Baptista, 2007; Tan *et al.*, 2021).

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 56–67.

La notion de défigement linguistique est intimement liée à la notion de figement. Le défigement est un phénomène qui vient briser une SF en lui retirant son caractère monolithique à la suite d'une ou plusieurs transformations linguistiques (Eline & Zhu, 2014). (Gross, 1996) va jusqu'à considérer le défigement comme un critère de reconnaissance du figement. À notre connaissance, contrairement aux SF, il n'existe pas de travaux de recherches s'intéressant spécifiquement à la reconnaissance automatique de séquences défigées (SD) à l'exception peut être de travaux sur la production de jeu de mots, énoncés que l'on peut voir comme des cas particuliers de défigements linguistiques (Valitutti *et al.*, 2013). Pourtant, ceci serait d'un intérêt triple d'un point de vue linguistique : (I) caractériser le figement d'expressions par leur productivité en termes de SD ; (II) détecter l'apparition de figements en « en temps réel » et (III) étudier les processus permettant aux locuteurs humains de reconnaître ces défigements. Nous faisons l'hypothèse qu'il est possible d'exploiter une méthodologie automatique permettant à des experts linguistiques d'explorer des corpus de manière non-supervisée afin d'opérer une veille informatisée d'un grand intérêt pour des chercheurs en linguistique.

Cet article est organisé de la façon suivante : dans la section 2 nous étudions les notions de figement et de défigement afin d'identifier les propriétés permettant de reconnaître automatiquement ces défigements à l'aide d'outils informatiques, dans la section 3 nous composons un corpus de tweets avec des défigements candidats, puis dans les sections 4 et 5 nous proposons des observations qualitatives et quantitatives sur les séquences défigées que nous avons extraites avant de proposer quelques perspectives de ce travail dans la section 6.

2 Propriétés linguistiques du défigement

Intrication des concepts de figement et de défigement Le figement linguistique se définit par trois critères principaux, issus de l'abondante littérature sur le sujet (Gross, 1982; Rommers *et al.*, 2013; Molinaro & Carreiras, 2010; Lamiroy, 2008; Mejri, 2005) :

non-compositionnalité du sens : on ne peut pas réduire le sens d'un figement au sens de chaque unité le composant, un sens global émerge ;

non-modifiabilité de la structure : l'ordre des mots composant le figement reste identique ;

non-substituabilité des termes : on ne peut pas remplacer l'un des termes du figement, même par un synonyme.

Le défigement vient « briser » un figement, les trois critères décrivant le figement permettent donc de détecter qu'un défigement se manifeste : « toute atteinte à la fixité formelle et à la globalité sémantique d'une SF serait considérée comme un défigement » (Mejri, 2009). Les études sur la notion de défigement s'accordent à dire que la reconnaissance d'un défigement implique au préalable la reconnaissance par le locuteur du figement dont il est issu (Fiala & Habert, 1989; Eline & Zhu, 2014) et que seul ce qui est figé se défige (Greciano, 1985). Pour caractériser chaque défigement, il faut par conséquent pouvoir identifier le figement à partir duquel ce défigement se forme. Une première piste pour notre travail de reconnaissance de SD est de créer une ressource de SF afin d'orienter les recherches et guider la reconnaissance. Selon (Eline & Zhu, 2014), au contraire du figement, le défigement est motivé, on peut donc considérer que l'on a une « remotivation » du figement, il faut que sa formation soit justifiée. Ce critère permet notamment de différencier l'énoncé fautif, par exemple une SF mal reconstituée par un locuteur [1], d'un réel défigement. Ce critère étant difficile à caractériser automatiquement, nous considérons ici que cette distinction serait réalisée par un expert linguiste en aval de la chaîne de traitement automatique. (Eline & Zhu, 2014) précisent également

1. Par exemple dire « être sur un même pied d'égalité » au lieu de « être sur un pied d'égalité »

que « toutes les séquences figées sont défigeables, mais toutes n'ont pas la même possibilité de défigement ». Les auteurs ajoutent que « plus la construction d'une SF est rare, plus la SF est apte à un défigement. ». Ces propriétés de productivité en défigements d'une SF et de fréquence nous seront utiles pour pouvoir mesurer parmi les défigements candidats ceux qui sont les plus intéressants en termes de créativité lexicale.

Degré de figement et défigement Toute séquence de mots possible dans la langue est en réalité plus ou moins figée, se plaçant dans un *continuum* allant de l'énoncé libre à la séquence figée (Gross, 1982; Mejri, 1998). Pour décrire ce phénomène, la littérature fait référence aux notions de degré de figement et d'opacité sémantique. Les semi-figements et quasi-figements (Mejri, 2005; François & Manguin, 2006) s'opposent ainsi aux figements absolus qui sont des figements non formellement contraints. Ces catégories de figement acceptent différentes variations et les SF y appartenant sont identifiées grâce à différents tests linguistiques tels que ceux effectués dans (Gross, 1982). (Nunberg *et al.*, 1994) précise qu'une SF peut posséder une interprétation littérale et une interprétation figée, pouvant être corrélées au sens compositionnel. En plus de variations formelles, nous pouvons donc observer des variations sémiques et un sens plus ou moins global en fonction des SF. Cette notion de degré de figement peut être problématique pour le TAL : une perte de la globalité du sens signifie-t-elle que nous sommes face à un défigement, ou à une SF avec un faible degré de figement ? Comment différencier une SF avec un degré de figement faible d'une SD ? Existe-t-il des cas ambigus ? Nous observons une relation entre degré de figement et défigement. Selon (Cusimano, 2015), une expression figée avec un degré de figement élevé a plus de chance de donner un défigement avec un degré de figement élevé.

Procédés de formation des défigements Nous identifions plusieurs procédés par lesquels des SD sont formées. (Galisson, 1993) parle de filiation phonique et de destructuration syntaxique. (Lecler, 2004) considère les défigement comme marqueurs dialogiques. (Eline & Zhu, 2014) font état d'un viol de la structure syntaxique, de la norme orthographique et d'une détérioration de la structure formelle des figements. Afin de visualiser ces changements avec des outils informatiques, nous devons récupérer des informations linguistiques sur plusieurs couches. Nous désirons analyser les couches syntaxiques, phonétiques et lexicales des expressions figées de notre corpus de figement et des possibles défigements de notre jeu de données. Tous les procédés de formation des défigements que nous venons de décrire illustrent des défigements marqués formellement : nous pouvons distinguer à la lecture le figement dont ils sont issus et les modifications apportées à la SF. Nous devons les dissocier des défigements non marqués formellement (Eline & Zhu, 2014). Ces défigements ne connaissent pas de changements formels par rapport aux figements dont ils sont issus. Leur statut de défigement est dû à des modifications de sens ou de prononciation par rapport à leur figement d'origine. Ils s'identifient donc exclusivement à l'aide du contexte dans lequel on les retrouve. Pour cette étude, nous nous intéressons exclusivement aux défigements analysables hors contexte. Afin de disposer d'un corpus de taille suffisante avec une créativité lexicale variée, nous avons choisi de nous intéresser aux réseaux sociaux numériques, et en l'espèce à TWITTER. Il s'agit de disposer de suffisamment d'exemples originaux pour avoir à la fois des « vrais positifs », de réels défigements, mais aussi des faux positifs, des contre-exemples qui vont permettre de questionner la qualité des algorithmes d'identification. La création de ce corpus est décrite dans la section suivante.

3 Identification de défigements candidats dans des tweets

Le processus que nous avons construit comporte trois étapes que nous décrivons ici : (I) l'extraction de tweets sources à partir de SF connues et le filtrage de ces tweets, (II) l'alignement des SF et des tweets et (III) l'isolement des segments communs constituant des défigements potentiels.

Extraction de tweets et filtrage. Pour collecter nos tweets, nous avons pris une base témoin de 217 expressions figées [2] appartenant à quatre catégories : (1) extraits ou slogans de publicité ; (2) citations politiques ou historiques ; (3) accroches pour des films de cinéma et (4) autres types de locutions. Nous avons cherché autant que possible à répartir équitablement les SF de ces catégories entre des expressions anciennes et plus récentes d'une part, et des expressions dont la connaissance est *a priori* répandue chez les locuteurs du français, ou au contraire plus confidentielle d'autre part. Ce critère est quelque peu subjectif, mais l'objectif est de pouvoir regarder en détails un nombre limité d'expressions. Nous donnons ci-dessous un exemple de SF pour chaque catégorie :

1. Tu pousses le bouchon un peu trop loin, Maurice.
2. On ne peut pas accueillir toute la misère du monde.
3. Dans l'espace personne ne vous entend crier.
4. Partir, c'est mourir un peu.

Les mots des SF sélectionnées sont utilisés pour faire des requêtes via l'API de TWITTER. Pour chaque requête, nous donnons une SF complète, ceci afin de ne pas trop biaiser le corpus d'étude en contraignant trop les résultats. Nous avons réalisé 3 collectes quotidiennes entre novembre 2020 et janvier 2023 inclus aboutissant à un total de 3 362 750 tweets extraits. Chaque tweet est associé à la SF qui nous a permis de l'extraire. Nous avons ensuite pré-filtré les tweets en ne conservant que ceux qui comportent au moins 50 % de mots en commun avec la SF d'origine. Au final, nous obtenons 99 244 tweets. Nous excluons ensuite les « figements » , c'est-à-dire les énoncés qui contiennent strictement l'expression figée recherchée. Le résultat de ce nouveau filtrage est représenté dans la Figure 1. Il résulte de tout cela un total de 56 687 tweets contenant potentiellement des défigements et 42 557 tweets contenant simplement un figement.

Encodage et alignement des séquences figées et de leurs défigements potentiels. Pour chaque tweet de notre corpus, nous calculons des alignements séquentiels pour visualiser les différences entre un défigement candidat et la SF associée. Ces alignements sont basés sur un découpage des tweets en mots et une analyse sur différentes couches d'informations linguistiques :

brute : tokenisation des formes de base du tweet et de la séquence figée ;

lemmatisée : recherche des formes canoniques pour traiter les micro-variations formelles ;

étiquetée syntaxiquement : pour rechercher la proximité de structure avec la SF ;

phonétisée : encodage en phonèmes pour valoriser les SD jouant sur la sonorité.

Afin de faciliter un futur traitement multilingue, nous avons utilisé SPACY pour les trois premières couches (tokenisation, lemmatisation et étiquetage) et EPITRAN [3] pour la phonétisation. Ensuite, nous alignons la SF et le tweet à l'aide de BIOPYTHON [4]. Cette librairie permet de calculer tous les alignements possibles entre deux séquences avec un alignement au token. Prenons l'expression

2. la liste complète est donnée sur le dépôt GitHub de notre projet https://github.com/JulienBez/DefigementTALN2023
3. https://pypi.org/project/epitran/
4. https://biopython.org/

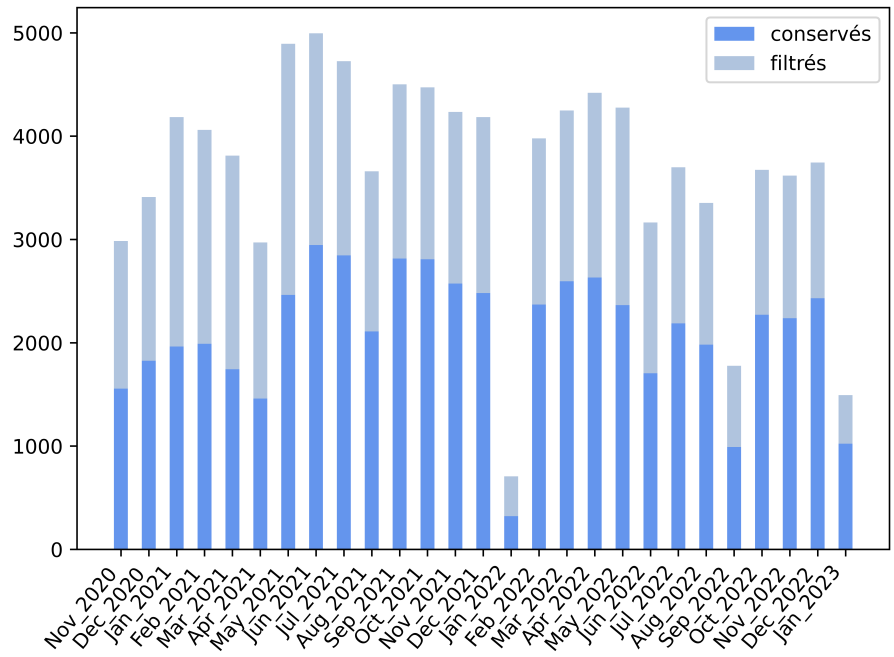


FIGURE 1 – Résultats de la récolte de tweets de novembre 2020 à janvier 2023 avant et après filtrage.

figée « Travailler plus pour gagner plus » et le défigement suivant, fréquent (voire banal) dans notre corpus : « Travailler plus pour gagner moins », l'alignement séquentiel obtenu serait :

Travailler	plus	pour	gagner	-	plus
Travailler	plus	pour	gagner	moins	-

Isolement des potentiels défigements. Les alignements contribuent à l'isolement d'une SD au sein d'un tweet, mais l'isolement n'est pas totalement réalisé puisque le tweet où le défigement est idéntifié peut contenir beaucoup plus de tokens que la SF d'origine. Les alignements sont commodes pour la lecture rapprochée des résultats mais ne permettent pas pour autant de délimiter précisément un défigement dans un tweet. L'étape suivante consiste donc à extraire les segments communs entre chaque tweet et son expression figée, c'est-à-dire d'extraire la plus grande séquence de mots commençant par le premier terme trouvé dans le tweet appartenant à l'expression figée et finissant par le dernier terme trouvé dans le tweet appartenant à cette expression. L'exemple ci-dessous représente un tweet de notre jeu de données, la SF qui a permis de sélectionner ce tweet et le segment commun identifié avec une distance d'édition au token de 2 (marqués en gras).

Tweet :	@user1 @user2 C bon ça ! !	Travailler	**moins**	pour	**bronzer**	plus	bye
SF :		Travailler	**plus**	pour	**gagner**	plus	
Défigement :		Travailler	**moins**	pour	**bronzer**	plus	

Les segments communs sont formés à partir de chaque alignement disponible pour un tweet (à raison d'un segment commun par alignement). Nous obtenons au moins un segment commun pour chaque couche d'information extraite du tweet (brute, lemmatisée, étiquetée et phonétisée). Dans le cas où, pour une même couche linguistique, nous avons plusieurs alignements, nous les conservons tous. Nous choisissons parmi ces alignements celui dont le segment commun est le plus proche de la SF [5]. Le Tableau 1 illustre de manière simplifiée ce que nous comparons entre une expression figée et le

5. Sans être strictement identique, puisque nous ne travaillons que sur les tweets ne comprenant pas la SF recherchée.

Couche	Séquence						Segment commun dans le tweet			Sim
BRUTE	Que	la	force	soit	avec	toi	**Que la force**	*(et la chance)*	**soit avec toi**	.76
LEMME	Que	le	force	être	avec	toi	**Que le force**	*(et le chance)*	**être avec toi**	.62
POS	SCONJ	DET	NOUN	VERB	ADP	PRON	*PRON* **DET NOUN**	*CCONJ DET NOUN*	*CCONJ* **ADP PRON**	.36
PHON	kə	la	fɔrs	swa	avək	twa	kə la fɔrs	(ɛ la ʃãsə)	swa avək twa	.76

TABLE 1 – Extraction du segment commun entre un tweet et une expression pour chaque couche linguistique (en gras les sous-séquences communes, en italiques les sous-séquences insérée).

segment commun extrait d'un tweet, nous y ajoutons la similarité cosinus calculée sur les vecteurs de bigrammes d'unités pour chaque couche (tokens, lemmes ...). Le segment commun permet de réduire le contenu du tweet à la sous-séquence constituant un potentiel défigement et donc de mieux mesurer la similarité que si nous gardions l'intégralité du tweet pour calcul. Pour les défigements simples (substitution ou insertion), le segment commun peut correspondre intégralement au potentiel défigement recherché, comme c'est le cas dans le Tableau 1.

4 Observations sur la qualité des défigements candidats

Cas de l'absence de défigement. Sans surprise, nous isolons des segments communs ne correspondant pas du tout à des défigements. Il s'agit de séquences dites libres. (1) et (2) en sont des exemples.

1. Tout ça pour travailler samedi prochain en plus yes. (SF : « travailler plus pour gagner plus »)
2. En train de regarder le plus beau travailler. (SF : « train de vie »)

Expressions sans trace de défigement. Nous n'utilisons pas les tweets contenant exactement la SF. Ces tweets ne font pas état d'un quelconque défigement. Nous notons deux situations ambiguës liées à la non prise en compte des tweets contenant l'expression recherchée mot pour mot : les défigements qui se prolongent dans le contexte du figement et les défigements non marqués formellement.

Défigement en milieu d'énoncé C'est le type de défigement le mieux identifié par notre méthodologie. Voici quelques exemples identifiés à partir des SF « Dans l'espace, personne ne vous entendra crier » et « Travailler plus pour gagner plus ».

3. Dans **la Beauce personne** ne vous entendra crier.
4. Travailler plus pour **redistribuer** plus.

Défigement en début/fin d'énoncé. Parfois, les modifications se situent en début ou fin d'expression, il est alors plus difficile de borner le segment commun. Ainsi, notre méthode ne permet pas d'isoler :

5. Dans l'espace, personne ne vous entendra crier **BONNE ANNÉE**.
6. Travailler plus pour **ne plus rien gagner**.

Expressions ouvertes sur leur contexte droit. Elles ont la spécificité d'être ouvertes sur leur contexte droit. Nous avons ainsi pour les SF « Ce moment où ... » et « Je traverse la rue et ... » :

7. Ce moment où **tu prends conscience que tu ne mérites pas ça...**
8. je traverse la rue et **je te trouve un boulot**.
9. Moi je traverse la rue et **je t'en gagne une de médaille d'or**.

Pour le moment, nous n'appliquons pas de traitements particuliers à ce type d'expressions. Nous envisageons cependant de préciser que pour ces expressions, il nous faut récupérer tout le contexte droit jusqu'à la première marque de fin de phrase (un point) ou même jusqu'à la fin du tweet.

5 Tri des défigements candidats par mesures de similarité

Afin d'évaluer si une approche par mesures de similarités permet de détecter des défigements, nous créons plusieurs classements des segments communs. Un classement est créé pour chaque mesure de similarité et pour chaque couche d'information linguistique. Nous cherchons à comparer systématiquement l'expression figée recherchée et les segments communs correspondants. Nous supposons qu'il doit exister un seuil, variable selon les expressions et les couches, à partir duquel les résultats renvoyés correspondent régulièrement à des SD. Nous testons différentes mesures de similarités (Cosinus, Dice, Hamming, Jaccard, Kulsinski, Matching et Rusell-rao). Nous allons observer plus en détails les défigements obtenus par couche linguistique pour la SF « Travailler plus pour gagner plus » (parmi 7 520 tweets contenant des défigements candidats). Nous prenons les dix défigements candidats les plus fréquents et nous les classons par mesure de similarité pour chaque couche. Les résultats obtenus avec la mesure de similarité cosinus sont présentés dans les Tableaux 2 à 5. Une SD remarquable a tendance à être plus utilisée selon (Cusimano, 2015), d'où notre choix de travailler sur les 10 candidats les plus fréquents.

Défigement potentiel	Sim	Freq
travailler plus pour gagner moins	0,80	364
travailler plus pour gagner pareil	0,80	10
travailler plus pour gagner autant	0,80	9
travailler plus pour payer plus	0,73	44
travailler plus sans gagner plus	0,73	21
travailler plus pour partager plus	0,73	10
travailler moins pour gagner plus	0,70	176
travailler plus pour vivre moins	0,60	16
travailler plus pour perdre moins	0,60	14
travailler moins et gagner plus	0,50	30

TABLE 2 – Couche brute

Défigement potentiel	Sim	Freq
travailler plus pour gagner moins	0,80	364
travailler plus pour gagner pareil	0,80	10
travailler plus pour gagner autant	0,80	9
travailler plus pour payer plus	0,73	44
travailler plus sans gagner plus	0,73	21
travailler plus pour partager plus	0,73	10
travailler moins pour gagner plus	0,70	176
travailler plus pour vivre moins	0,60	16
travailler plus pour perdre moins	0,60	14
travailler moins et gagner plus	0,50	30

TABLE 3 – Couche lemmatisée

Défigement potentiel	Sim	Freq
travailler plus pour gagner moins	1,0	438
travailler moins pour gagner plus	1,0	179
travailler plus pour payer plus	1,0	44
travailler plus sans gagner plus	1,0	21
travailler plus pour vivre moins	1,0	18
travailler plus pour perdre moins	1,0	14
travailler plus pour gagner autant	1,0	12
gagner plus sans travailler plus	1,0	11
travailler plus pour produire plus	1,0	10
travailler plus pour mourir plus	1,0	7

TABLE 4 – Couche étiquetée

Défigement potentiel	Sim	Freq
travailler plus pour gagner moins	0,80	364
travailler plus pour gagner pareil	0,80	10
travailler plus pour gagner autant	0,80	9
travailler plus pour payer plus	0,73	44
travailler plus sans gagner plus	0,73	21
travailler plus pour partager plus	0,73	10
travailler moins pour gagner plus	0,70	176
travailler plus pour vivre moins	0,60	16
travailler plus pour perdre moins	0,60	14
travailler moins et gagner plus	0,57	30

TABLE 5 – Couche phonétisée

Nous remarquons pour les couches brute, lemmatisée et phonétisée des résultats identiques avec sept défigements pour des mesures de similarités allant de 0,7 à 0,8 et trois défigements pour des mesures de similarité allant de 0,6 à 0,5. Nous observons deux différences notables entre les résultats obtenus avec ces trois couches et les résultats obtenus à la couche étiquetée. La première est une différence de fréquence des segments communs. Cela peut s'expliquer par le nombre important d'alignements trouvés pour la couche étiquetée (34 008 alignements) par rapport aux autres couches (11 708 pour

les couches brute et lemmatisée, 11 710 pour la couche phonétisée). Nous rappelons que pour chaque tweet, un alignement est renvoyé par possibilité d'alignement identifiée avec la librairie BIOPYTHON et ce pour chaque couche linguistique étudiée. Comme les étiquettes morphosyntaxiques d'une SF peuvent être assez communes et se retrouver de multiples fois dans un tweet, il n'est pas surprenant que les alignements morphosyntaxiques soient multiples. La seconde différence concerne les mesures de similarité obtenues. Sont renvoyés avec une mesure de similarité égale à 1 tous les segments communs dont la structure morphosyntaxique correspond exactement à celle de la SF. Bien que les segments communs renvoyés dans le Tableau 4 correspondent tous à des défigements, nous retrouvons également des segments communs comme (10), (11) et (12) avec une même similarité de 1.

10. Fait tout pour agir maintenant
11. Je fais comment pour travailler
12. Commence alors à décliner progressivement

Pour cette expression, la couche étiquetée ne s'avère pas efficace, par contre en observant les trois autres couches, nous observons qu'un seuil de similarité de 0,7 pour Cosinus et Dice permet de ne garder que des SD. Nous présentons dans le Tableau 6 les 20 résultats les plus fréquents obtenus avec la couche brute et une similarité cosinus entre 0,8 et 0,7 et nous y ajoutons les résultats obtenus avec les autres mesures de similarité pour chaque segment commun. Nous remarquons (lignes grisées) un ensemble de SD possédant les mêmes résultats d'une mesure de similarité à l'autre (0,73 pour la similarité cosinus). Au total, nous comptons 74 potentiels défigements avec les mêmes résultats pour chaque mesure de similarité pour notre SF. Sur ces 73 potentiels défigements, 65 sont des SD et 8 ne sont pas des SD. Nous divisons les séquences qui ne sont pas des SD en deux catégories : les SF avec un énoncé fautif (13 à 16) et les SD qui n'ont été capturées que partiellement (17 à 20).

13. travailler plus poyr gagner plus
14. travailler plus pour gagné plus
15. travailler plus pour gagniez plus
16. travailler plus pour gagnez plus
17. travailler plus pour le plus
18. travailler plus pour la plus
19. travailler plus pour l ukraine plus
20. travailler plus pour être plus

Toutes les SD obtenues pour ces résultats sont globalement de la même forme : il s'agit de défigements par substitution d'un des termes de l'expression recherchée par un autre terme. Nous pourrions isoler un seuil de similarité dans lequel nous observerions toutes les occurrences de SD par substitution d'un terme. Cependant, ce seuil varie selon les expressions. On remarque des SD formées par substitution d'un terme avec des similarités supérieures et inférieures à celles que nous analysons, que nous indiquons en bleu dans le Tableau 6. Nous observons que la position des termes substitués a un impact sur les mesures de similarités renvoyées. Le seuil d'identification varie sans doute selon le procédé de formation du défigement. Notons que ces seuils ne garantissent pas l'absence de faux positifs. Il faudrait pouvoir quantifier ces faux positifs pour chaque seuil et établir une méthode permettant de les trier. Par exemple, pour les fautes d'orthographe, nous pouvons observer la couche lemmatisée : pour notre seuil, on éliminerait ainsi les faux positifs (13), (14) et (15), qui contiennent des formes mal orthographiées du verbe « gagner ». De même, si nous recherchons des défigements par substitution, il est très probable que les SD aient une similarité de 1 sur la couche morphosyntaxique avec nos mesures de similarité. On exclut ainsi (17), (18), (19) et potentiellement (13) du fait de la faute d'orthographe, non répertoriée dans le lexique et non-étiquetée. En revanche, (20) n'est pas écarté avec ces filtrages.

Défigement potentiel					Cos	Dic	Ham	Jac	Kul	Mat	Rus	#
travaille	plus	pour	gagner	moin	,80	,82	,60	,70	,54	,70	,70	364
travaille	plus	pour	gagner	parei	,80	,82	,60	,70	,54	,70	,70	10
travaille	plus	pour	gagner	autar	,80	,82	,60	,70	,54	,70	,70	9
travaille	plus	pour	travailler	plus	,78	,71	,33	,56	,38	,56	,56	5
travailler	plus	pour	gagner un peu	plus	,78	,70	,54	,54	,37	,54	,54	3
travaille	plus	pour	payer	plus	,73	,62	,45	,45	,29	,45	,45	44
travaille	plus	sans	gagner	plus	,73	,62	,45	,45	,29	,45	,45	21
travaille	plus	pour	partager	plus	,73	,62	,45	,45	,29	,45	,45	10
travaille	plus	pour	produire	plus	,73	,62	,45	,45	,29	,45	,45	9
travaille	plus	et	gagner	plus	,73	,62	,45	,45	,29	,45	,45	9
travaille	plus	pour	être	plus	,73	,62	,45	,45	,29	,45	,45	7
travaille	plus	pour	mourir	plus	,73	,62	,45	,45	,29	,45	,45	7
travaille	plus	pour	crever	plus	,73	,62	,45	,45	,29	,45	,45	7
travaille	plus	pour	donner	plus	,73	,62	,45	,45	,29	,45	,45	6
travaille	plus	pour	perdre	plus	,73	,62	,45	,45	,29	,45	,45	5
travaille	plus	pour	avoir	plus	,73	,62	,45	,45	,29	,45	,45	4
travaille	plus	pour	faire	plus	,73	,62	,45	,45	,29	,45	,45	4
travaille	moins	pour	gagner	plus	,70	,71	,45	,55	,38	,55	,55	176
travaille	autant	pour	gagner	plus	,70	,71	,45	,55	,38	,55	,55	5
travailler	à l étranger	pour	gagner	plus	,70	,70	,45	,55	,38	,55	,55	2

TABLE 6 – Résultats obtenus avec la SF « Travailler plus pour gagner plus » pour chaque mesure de similarité, classés par similarité cosinus décroissante.

Nous réalisons la même expérience avec une nouvelle SF : « Que la force soit avec toi » (1 523 occurrences). Nous obtenons le Tableau 7. Nous retrouvons en gris les potentiels défigements dont la mesure de similarité cosinus est égale à 0,73. Là encore, nous retrouvons pour ces résultats des défigements formés par la substitution d'un des termes de la SF. Nous remarquons tout de même des différences entre les résultats des mesures de similarités d'une SF à l'autre (sauf pour les similarités Cosinus et Kulsinski).

Les SD « que la force et le courage soit avec toi » et « que le pouvoir de la force soit avec toi » ont bien une mesure de similarité cosinus égale à 0,73 mais leur procédé de formation n'est pas la substitution d'un des termes de l'expression recherchée. Nous remarquons des résultats différents pour toutes les autres mesures de similarités pour ces deux SD, ce qui permet de les isoler des autres SD avec une distance cosinus de 0,73 qui sont bien formées par substitution d'un terme.

Toujours dans le Tableau 7, nous indiquons en bleu un second seuil à partir duquel nous observons des SD formées par insertion d'un, deux ou trois mots au sein de l'expression recherchée. Il est donc bien possible d'observer des types de défigements différents en fonction du seuil que nous analysons. À ce stade, nous trouvons ces seuils par une lecture des résultats obtenus pour deux expressions. Un de nos prochains objectifs sera d'identifier automatiquement les seuils permettant d'isoler uniquement des SD. La finalité de ce travail nous permettra de savoir si des seuils de mesure de similarité peuvent représenter plusieurs types de défigement, comme c'est le cas avec les exemples que nous avons traités.

Potentiel défigement			Cos	Dic	Ham	Jac	Kul	Mat	Rus	#
que	la force de dieu	soit avec toi	,78	,77	,62	,62	,45	,62	,62	9
que	la force du café	soit avec toi	,78	,77	,62	,62	,45	,62	,62	6
que	la force de guérir	soit avec toi	,78	,77	,62	,62	,45	,62	,62	6
que	la force et l amour	soit avec toi	,78	,77	,62	,62	,45	,62	,62	4
que	la force du vent	soit avec toi	,78	,77	,62	,62	,45	,62	,62	3
que	la force du tigre	soit avec toi	,78	,77	,62	,62	,45	,62	,62	3
que	la force du dragon	soit avec toi	,78	,77	,62	,62	,45	,62	,62	3
que	la force	soit avec moi et toi	,78	,77	,62	,62	,45	,62	,62	2
que	la force of god	soit avec toi	,78	,77	,62	,62	,45	,62	,62	2
que	la force et la patience	soit avec toi	,76	,74	,53	,59	,42	,59	,59	4
que	là force	soit avec toi	,73	,73	,57	,57	,4	,57	,57	9
que	la paix	soit avec toi	,73	,73	,57	,57	,4	,57	,57	6
que	la force	soit en toi	,73	,73	,57	,57	,4	,57	,57	6
que	la force et le courage	soit avec toi	,73	,71	,56	,56	,38	,56	,56	6
que	le pouvoir de la force	soit avec toi	,73	,71	,56	,56	,38	,56	,56	3
que	le force	soit avec toi	,73	,73	,57	,57	,4	,57	,57	3
que	la santé	soit avec toi	,73	,73	,57	,57	,4	,57	,57	2
que	la réussite	soit avec toi	,73	,73	,57	,57	,4	,57	,57	2
que	la force	ne soit pas avec toi	,70	,69	,53	,53	,36	,53	,53	6
que	la force	ne soit jamais avec toi	,70	,69	,53	,53	,36	,53	,53	2

TABLE 7 – Résultats obtenus avec la SF « Que la force soit avec toi » pour chaque similarité, classés selon la mesure de similarité cosinus.

6 Conclusion et perspectives

Dans cet article, nous nous sommes intéressés à un cas particulier de créativité lexicale : le défigement linguistique. Nous avons construit un corpus de tweets sur lequel nous avons appliqué une méthode de reconnaissance de séquences défigées (SD) fondée sur des mesures de similarité. Nous avons exploité des mesures de similarité pour identifier des SD dans ce corpus de tweets. Nous avons utilisé différentes couches d'analyse linguistique pour parvenir à une reconnaissance de ces SD en nous limitant à une approche des défigements interprétables hors contexte. Dans le futur, nous envisageons d'exploiter plus avant la notion de « remotivation » des SD telle que proposée par (Eline & Zhu, 2014) qui pourrait permettre de différencier les SD volontaires des énoncés fautifs et d'interroger la relation entre degré de figement et défigement. Nous estimons qu'une mesure de similarité combinant les couches linguistiques brutes, étiquetée et lemmatisée permet une meilleure reconnaissance de défigements. Il nous reste à déterminer si la couche phonétique peut, elle aussi, se révéler utile dans notre tâche et à exploiter également une couche syllabique. Les résultats obtenus avec les mesures de similarité nous montrent qu'il est possible de créer une classification des défigements. On observe des seuils de similarité permettant de regrouper des SD formées par les mêmes procédés linguistiques. Nous nous demandons si ces seuils devront être « fixés » globalement ou pourront être calculés pour chaque expression, ou pour chaque procédé de construction. Déterminer automatiquement ces seuils, en exploitant des modèles de langue contextuels est une autre voie que nous comptons explorer.

Références

BAPTISTA J., CORREIA A. & FERNANDES G. (2004). Frozen Sentences of Portuguese : Formal Descriptions for NLP. In T. TANAKA, A. VILLAVICENCIO, F. BOND & A. KORHONEN, Éds., *ACL Workshop on Multiword Expressions : Integrating Processing*, p. 72–79, Barcelona, Spain. HAL : hal-01025937.

CATALÀ D. & BAPTISTA J. (2007). Spanish adverbial frozen expressions. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, p. 33–40.

CUSIMANO C. (2015). Figement de séquences défigées. *Pratiques*, (159-160), 69 78. DOI : 10.4000/pratiques.2833.

ELINE J. & ZHU L. (2014). Défigement et inférence - cas d'études du Canard enchaîné. *SHS Web of Conferences*, **8**, 681 695. DOI : 10.1051/shsconf/20140801235.

FIALA P. & HABERT B. (1989). La langue de bois en éclat : les défigements dans les titres de presse quotidienne française. *Mots. Les langages du politique*, **21**(1), 83–99. DOI : 10.3406/mots.1989.1504.

FORT K., GUILLAUME B., CONSTANT M., LEFÈBVRE N. & PILATTE Y.-A. (2018). "Fingers in the Nose" : Evaluating Speakers' Identification of Multi-Word Expressions Using a Slightly Gamified Crowdsourcing Platform. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, p. 207–213, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

FORT K., GUILLAUME B., PILATTE Y.-A., CONSTANT M. & LEFÈBVRE N. (2020). Rigor Mortis : Annotating MWEs with a Gamified Platform. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 4395–4401, Marseille, France : European Language Resources Association.

FRANÇOIS J. & MANGUIN J.-L. (2006). Dispute théologique, discussion oiseuse et conversation téléphonique : les collocations adjectivo-nominales au cœur du débat. *Langue française*, **150**(2), 50–65. DOI : 10.3917/lf.150.0050.

GALISSON R. (1993). Les palimpsestes verbaux : des révélateurs culturels remarquables, mais peu remarqués. *Repères. Recherches en didactique du français langue maternelle*, **8**(1), 41–62. DOI : 10.3406/reper.1993.2091.

GRECIANO (1985). Gréciano Gertrud, Signification et dénotation en Allemand. La sémantique des expressions idiomatiques, Paris, Klincksieck, 1983. *L'information grammaticale*, **24**(1), 47–48.

GROSS G. (1996). *Les expressions figées en français. Noms composés et autres locutions - Gaston Gross*. OPHRYS.

GROSS M. (1982). Une classification des phrases « figées » du français. *Revue québécoise de linguistique*, **11**(2), 151. DOI : 10.7202/602492ar.

LAMIROY B. (2008). Le figement : à la recherche d'une définition. *ZFSL, Zeitschrift für französische Sprache und Literatur*, **36**, 85–99.

LECLER A. (2004). Blague à part, peut-on traiter la question du défigement en termes dialogiques ? *Cahiers de praxématique*, (43), 81–106. DOI : 10.4000/praxematique.1807.

LECLÈRE C. (2000). Expressions figées dans la francophonie : le projet bfqs.

MEJRI S. (1998). La conceptualisation dans les séquences figées. *L'information grammaticale*, **2**(1), 41–48. DOI : 10.3406/igram.1998.3699.

MEJRI S. (2005). Figement absolu ou relatif : la notion de degré de figement. *Linx. Revue des linguistes de l'université Paris X Nanterre*, (53), 183–196. DOI : 10.4000/linx.283.

MEJRI S. (2009). Figement, défigement et traduction. Problématique théorique. *Pratiques*, p. 153.

MOLINARO N. & CARREIRAS M. (2010). Electrophysiological evidence of interaction between contextual expectation and semantic integration during the processing of collocations. *Biological Psychology*, **83**, 176–190. DOI : 10.1016/j.biopsycho.2009.12.006.

NUNBERG G., SAG I. A. & WASOW T. (1994). Idioms. *Language*, **70**(3), 491–538. DOI : 10.2307/416483.

ROMMERS J., DIJKSTRA T. & BASTIAANSEN M. (2013). Context-dependent Semantic Processing in the Human Brain : Evidence from Idiom Comprehension. *Journal of Cognitive Neuroscience*, **25**(5), 762–776. DOI : 10.1162/jocn_a_00337.

TAN M., JIANG J. & DAI B. T. (2021). A bert-based two-stage model for chinese chengyu recommendation. *Transactions on Asian and Low-Resource Language Information Processing*, **20**(6), 1–18.

VALITUTTI A., TOIVONEN H., DOUCET A. & TOIVANEN J. M. (2013). "let everything turn well in your wife" : generation of adult humor using lexical constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 243–248.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Tri-apprentissage génératif : génération de données pour de la reconnaissance d'entitées nommées semi-supervisé

Untel Trucmuche[1, 2] Unetelle Machinchose[1, 3]
(1) Lab, adresse, CP Ville, Pays
(2) Lab, adresse, CP Ville, Pays
(3) Lab, adresse, CP Ville, Pays
utrucmuche@lab.fr, umachinchose@adresse-academique.be

RÉSUMÉ

Le développement de solutions de traitement automatique de la langue pour de nouvelles tâches nécessite des données, dont l'obtention est coûteuse. L'accès aux données peut être limité en raison de la nature sensible des données. La plupart des travaux récents ont exploité de grands modèles pré-entraînés pour initialiser des versions spécialisées de ceux-ci. La spécialisation d'un tel modèle nécessite toujours une quantité élevée de données étiquetées spécifiques à la tâche cible. Nous utilisons l'apprentissage semi-supervisé pour entraîner des modèles dans un contexte où le nombre d'exemples étiquetés est limité et le nombre de données non étiquetées est nul. Nous étudions plusieurs méthodes pour générer le corpus non étiqueté nécessaire à l'utilisation de l'apprentissage semi-supervisé. Nous introduisons les méthodes de génération entre les épisodes d'entraînement et utilisons les modèles entraînés pour filtrer les exemples générés. Nous testons cette génération avec le tri-apprentissage et l'auto-apprentissage sur des corpus Anglais et Français.

ABSTRACT

Generative Tri-training : Semi-Supervised NER with On-the-fly Generation of Unlabeled Data.

Developing Natural Language Processing solutions to new tasks or domains requires data, which is costly to collect and annotate. For particular domains, access to data is limited due to the sensitive nature of the data. These past years, most works have leveraged large general pre-trained models to initialize specialized versions of these. The specialization of such a model still requires a relatively high quantity of labeled data specific to the target task. We use semi-supervised learning to train models in a context where the number of labeled examples is limited, and the number of unlabeled data is zero. Since semi-supervised learning requires unlabeled examples, we study methods of generating the necessary unlabeled corpus. We introduce the generation methods between the training episodes and use the models trained to filter the generated examples. We test this on-the-fly generation with tri-training and self-training with corpora in English and French.

MOTS-CLÉS : REN, génération, faible ressources, semi-supervision.

KEYWORDS: NER, generation, low resource, semi-supervision.

1 Introduction

La reconnaissance d'entités nommées (REN) est une tâche d'étiquetage de séquences. L'entraînement de modèles de REN nécessite généralement une quantité importante de données étiquetées. L'accès

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 68–79.

aux données, étiquetées ou non, est compliqué pour certaines langues ou certains domaines. Par exemple, obtenir des données médicales prend des années car l'obtention de données diffusables et l'annotation par des spécialistes est un processus lent. Par conséquent, l'apprentissage dans un environnement à faibles ressources est un sujet en plein essor. Le but de cet article est de proposer une méthode permettant de réduire la quantité de données naturelles annotées nécessaires pour entraîner des modèles de REN.

Les techniques d'apprentissage par transfert (Ruder, 2019) sont un moyen d'utiliser les connaissances acquises depuis d'autres données pour améliorer les performances des modèles. L'apprentissage semi-supervisé est un autre paradigme d'apprentissage qui aborde le problème des ressources limitées (Van Engelen & Hoos, 2020). Dans ce paradigme, le contexte de faibles ressources fait généralement référence à une faible quantité de données étiquetées mais à une grande quantité de données non étiquetées disponibles. Contrairement à l'apprentissage semi-supervisé, le problème que nous cherchons à résoudre est de n'avoir qu'une petite quantité de données étiquetées et aucune donnée non étiquetée. Nous utilisons de grands modèles de langue génériques sans réglage fin pour générer les données non étiquetées utilisées dans l'apprentissage semi-supervisé. Nous n'affinons pas nos modèles de génération car nous ne disposons pas de suffisamment de données.

Nous évaluons plusieurs méthodes de génération utilisant des modèles de langue pré-entraînés. Premièrement, nous employons deux méthodes de génération fondées sur la modélisation classique de gauche à droite : la génération de la phrase suivante et la complétion de phrases. Ensuite, nous utilisons deux méthodes de modélisation séquence à séquence pour remplacer le contexte ou les mentions dans les phrases étiquetées. Nous évaluons notre méthode sur deux corpus bien connus, CoNLL (Sang & De Meulder, 2003) et I2B2 (Uzuner *et al.*, 2011). Nos principales contributions sont les suivantes [1] :

— L'*algorithme de tri-apprentissage génératif*, un algorithme qui ajoute la génération et la sélection des exemples non étiquetés à chaque épisode de l'algorithme de tri-apprentissage.
— Une analyse complète des méthodes de génération multiples pour l'algorithme de tri-apprentissage générative.

2 Contexte

L'apprentissage d'un modèle de REN nécessite une grande quantité de données étiquetées. L'augmentation a été utilisée pour améliorer les performances des modèles de traitement automatique de la langue. Des techniques telles que la traduction circulaire (Sennrich *et al.*, 2016), l'augmentation EDA (Wei & Zou, 2019) ou la génération de paraphrases en utilisant BART (Dopierre *et al.*, 2021) ont été utilisées pour améliorer la classification de phrases. Cependant, la paraphrase utilisant la rétro-traduction sur des données médicales pour une tâche d'étiquetage n'est pas efficace (Neuraz *et al.*, 2018). Par conséquent, nous ne l'utiliserons pas sous cette forme, mais l'utilisation de modèle de langue se rapproche de cette pratique. Une méthode utilisant des modèles de langue pour augmenter les données, DAGA (Ding *et al.*, 2020), s'est avérée efficace dans un contexte d'apprentissage supervisé et semi-supervisé. Cette méthode utilise les données d'apprentissage pour entraîner un BiLSTM afin de générer des données étiquetées ou non étiquetées. Nous avons choisi d'utiliser des modèles de langue pré-entraînés.

L'apprentissage semi-supervisé est un paradigme d'apprentissage visant à améliorer les performances

1. Le code sera disponible sur un dépôt public après publication.

des modèles entraînés en ajoutant des exemples non étiquetés à l'ensemble d'apprentissage (Van Engelen & Hoos, 2020). Ce paradigme comporte plusieurs branches qui dépendent de la manière dont les données non étiquetées sont utilisées. L'utilisation de pseudo-étiquettes à différents stades de l'entraînement a fonctionné pour la REN (Wang *et al.*, 2021b). Les modèles aux stades d'apprentissage précédents génèrent des pseudo-étiquettes pour les étapes suivantes. Ces algorithmes dépendent du nombre de modèles entraînés et des modèles utilisés pour générer les pseudo-étiquettes. L'algorithme le plus simple est l'auto-apprentissage (Yarowsky, 1995), qui entraîne un modèle et l'utilise pour créer les pseudo-étiquettes. D'autres méthodes utilisant des ensembles de modèles ont été créées pour réduire les biais induits par la génération des pseudo-étiquettes par le même modèle. Le co-training (Blum & Mitchell, 1998) est une méthode d'entraînement de deux modèles dans laquelle chaque modèle génère les pseudo-étiquettes de l'autre. Une généralisation de cette méthode existe dans laquelle un ensemble de n modèles est entraîné, et les pseudo-étiquettes pour un modèle m_j sont produites en utilisant un système de vote à travers les n autres modèles. L'algorithme semi-supervisé que nous utilisons est une variante utilisant trois modèles appelée tri-apprentissage (Zhou & Li, 2005). Des résultats positifs existent avec cette méthode sur la tâche d'extraction de concepts cliniques. Une version optimisée pour les données de l'algorithme de tri-apprentissage est le tri-apprentissage avec désaccord (Søgaard, 2010). Cette version réduit la quantité de données nécessaires en ajoutant uniquement les données pseudo-étiquetées au jeu d'entraînement d'un modèle quand celui-ci est en désaccord avec les deux autres. Nous n'avons pas implémenté cette version du tri-apprentissage car nous voulons conserver autant de données générées pertinentes que possible. Nous proposons un algorithme de tri-apprentissage génératif, qui utilise le système de vote pour écarter les échantillons nouvellement générés sur lesquels il n'y a pas d'accord.

Nous utilisons des modèles de langue pour générer de nouvelles données non étiquetées nécessaires à l'apprentissage semi-supervisé. Nous utilisons les modèles GPT-2 (Radford *et al.*, 2019) et T5 (Raffel *et al.*, 2020) dans leur version à 1M paramètres. Ces données servent à entraîner nos étiqueteurs, qui utilisent des modèles BERT (Devlin *et al.*, 2018) pré-entraînés comme base de leur architecture.

3 Tri-apprentissage génératif

Tri-apprentissage Le tri-apprentissage (Zhou & Li, 2005) est un algorithme d'apprentissage semi-supervisé pour entraîner un ensemble de trois modèles. Chaque modèle est entraîné sur des données non étiquetées qui reçoivent des pseudo-étiquettes des deux autres modèles. Les phrases pseudo-étiquetées sont utilisées lorsque les deux modèles parviennent à un accord, ce qui permet à la fois l'ajout de pseudo-étiquettes et un filtrage des phrases non-adaptées. La génération d'un ensemble de données non étiquetées de taille fixe pour l'apprentissage semi-supervisé semble arbitraire. Il n'y a aucune garantie que les modèles utiliseront les données pour apprendre car il n'y a aucune garantie que les modèles parviendront à un accord sur les pseudo-étiquettes. En pratique, certaines des données générées ne sont jamais utilisées. Pour résoudre ce problème, nous avons conçu l'algorithme de tri-apprentissage génératif qui se sert des qualités de filtrage du tri-apprentissage pour sélectionner les phrases issues de la génération au fil des épisodes.

Pré-entraînement Le tri-apprentissage nécessite une étape d'entraînement pour initialiser les étiqueteurs afin qu'ils soient capables de produire des pseudo-étiquettes. Nous appelons cette étape l'étape de pré-entraînement. Elle est représentée entre les lignes 1 et 4 de l'Algorithme 1. L'étape

Algorithme 1 : Tri-apprentissage génératif

Entrées : S_n le sous-ensemble de données annotées, g la méthode de génération

1 **pour** $i \in [\![1\ ;\ 3]\!]$ **faire**
2 $\quad m_i^{-2} \leftarrow entrainement(echantillonnage(S_n), BERT)$
3 $\quad m_i^{-1} \leftarrow entrainement(S_n, m_i^{-2})$
4 **fin**
5 $t \leftarrow 0$
6 $U^0, L_1^{-1}, L_2^{-1}, L_3^{-1} \leftarrow \emptyset$
7 **tant que** *un* m_i *apprend toujours* **faire**
8 $\quad L_1^t, L_2^t, L_3^t \leftarrow generation\left(S_n \cup \bigcup_{i=1}^3 L_i^{t-1}, m_{i,i\in[\![1\ ;\ 3]\!]}^{t-1}, g\right)$
9 $\quad U^{t+1} \leftarrow U^t \cup enlever\_etiquettes\left(\bigcup_{i=1}^3 L_i^t\right)$
10 $\quad$ **pour** $i \in [\![1\ ;\ 3]\!]$ **faire**
11 $\quad\quad j, k \leftarrow [\![1\ ;\ 3]\!] - \{i\}$
12 $\quad\quad$ **pour** $x \in U^t$ **faire**
13 $\quad\quad\quad$ **si** $m_j^{t-1}(x) = m_k^{t-1}(x)$ *and* $(m_i^{t-1}(x) \neq m_j^{t-1}(x)$ *or* $drop(p > .5))$ **alors**
14 $\quad\quad\quad\quad L_i^t \leftarrow L_i^t \cup \{(x, m_j^{t-1}(x))\}$
15 $\quad\quad\quad$ **fin**
16 $\quad\quad$ **fin**
17 $\quad$ **fin**
18 $\quad$ **pour** $i \in [\![1\ ;\ 3]\!]$ **faire**
19 $\quad\quad$ **si** m_i *apprend toujours* **alors**
20 $\quad\quad\quad m_i^t \leftarrow entrainement(S_n \cup L_i^t, m_i^{t-1})$
21 $\quad\quad$ **fin**
22 $\quad$ **fin**
23 $\quad t \leftarrow t + 1$
24 **fin**

de pré-entraînement est effectuée sur des sous-ensembles de données échantillonnés avec remise à partir de l'ensemble étiqueté S_n (Ruder & Plank, 2018). Ces sous-ensembles font **la même taille que S_n.** Cependant, nous avons constaté que l'ajout d'un épisode d'apprentissage sur l'ensemble étiqueté complet S_n après l'ensemble échantillonné améliore considérablement les performances du tri-apprentissage. Cet ajout se trouve à la ligne 3 de l'Algorithme 1.

Tri-apprentissage génératif L'algorithme 1 et la figure 1 reflètent les modifications apportées à l'algorithme de tri-apprentissage. Avec le tri-apprentissage, nous entraînons un ensemble de trois modèles $m_i, i \in [\![1\ ;\ 3]\!]$. L'entraînement est divisé en épisodes au cours desquels nous entraînons les trois modèles. Pour un modèle m_i donné, l'entraînement épisodique s'arrête lorsque le score du modèle sur l'ensemble de validation est inférieur à celui de l'épisode précédent. Ce processus est décrit dans la Figure 2. Au début de chaque épisode (ligne 8 de l'algorithme 1), nous générons de nouveaux sous-ensembles de données pseudo-étiquetées L_i^t à partir des données annotées et pseudo-annotées de l'épisode précédent. Nous ajoutons les données générées sans annotation aux données non-annotées disponibles pour l'étape suivante ligne 9. Ces sous-ensembles sont générés à partir des ensembles étiquetés et précédemment pseudo-étiquetés. Nous ajoutons les exemples nouvellement générés L_i^t

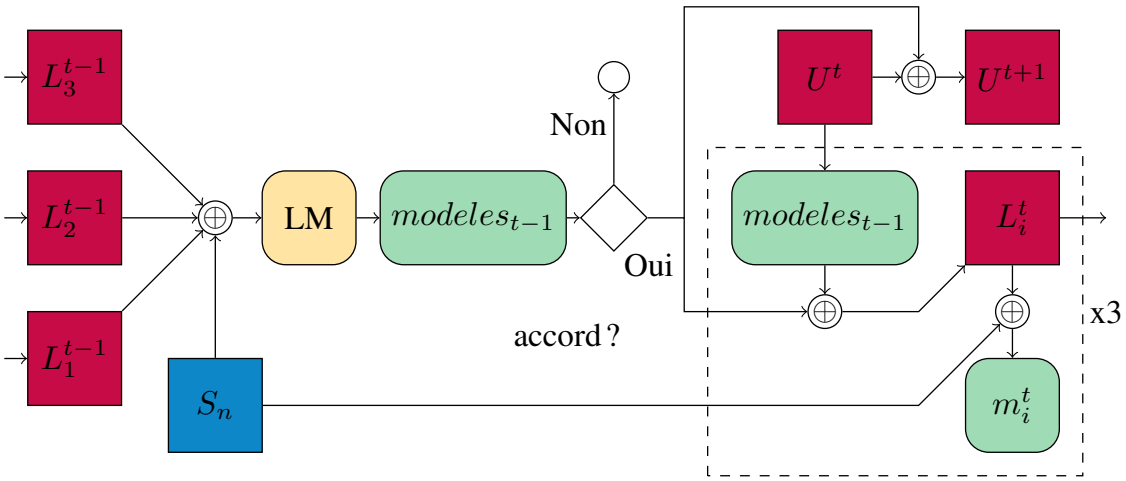


FIGURE 1 – Étape t du tri-apprentissage génératif. En rectangles les données, et avec les coins arrondis les modèles et ensembles de modèles.

sans leurs étiquettes aux exemples précédemment générés U^t pour l'épisode suivant. Chacun de ces ensembles pseudo-étiquetés L_i^t est ensuite augmenté à l'aide d'exemples précédemment générés U^t avec le mécanisme classique de tri-apprentissage (lignes 10 à 17 de Alg. 1). Enfin, entre les lignes 18 et 22, nous entraînons les modèles qui nécessitent encore un entraînement. Cet algorithme résout le problème des données inutilisées en rejetant les exemples fraîchement générés sur lesquels il n'y a pas d'accord, comme le montre la Figure 1. Nous utilisons une modification du désaccord avec une chance sur deux pour chaque modèle de conserver les exemples pseudo-étiquetés sur lesquels il y a un accord complet, comme le montrent les lignes 13 et 14. La motivation derrière ce changement est de conserver en parti les exemples vus par les modèles pour éviter l'oubli, tout en bénéficiant d'une quantité de données plus faible pour améliorer la vitesse d'apprentissage.

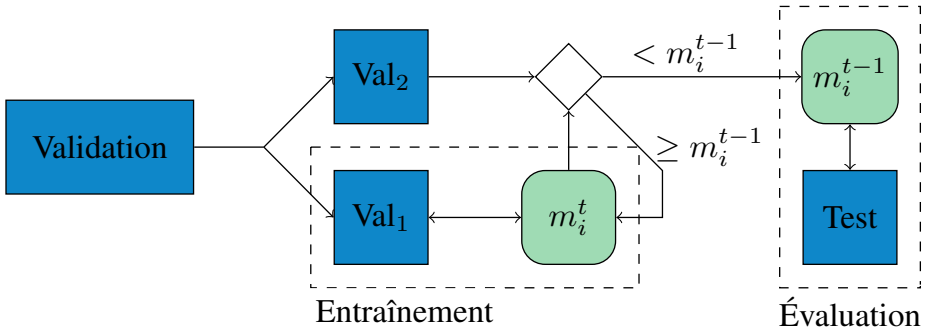


FIGURE 2 – Mécanisme de validation pour un modèle m_i. Nous découpons le jeu de validation du corpus naturel en deux, une partie servant durant l'entraînement, et une autre servant à comparer entre les épisodes.

Pour le tri-apprentissage, et l'aspect épisodique en général, nous avons besoin de deux ensembles de données de validation pour évaluer notre entraînement. Nous divisons l'ensemble de validation de nos corpus en deux. La première moitié sert d'ensemble de validation pour l'entraînement des modèles pendant l'épisode. La seconde moitié est utilisée pour comparer les modèles entre les épisodes et définir le moment où un modèle doit arrêter son apprentissage. Ce mécanisme est décrit dans la Figure 2.

4 Méthodes de génération

Les méthodes pour générer de nouveaux échantillons non étiquetés sont au cœur de cette étude. La modélisation de la langue est l'outil que nous utiliserons pour générer de nouveaux exemples. Il existe plusieurs façons de faire de la modélisation de la langue. La façon la plus traditionnelle de faire de la modélisation de la langue est de prédire les tokens de gauche à droite. Ces modèles utilisent le contexte à gauche pour prédire les tokens suivants à droite. Nous utiliserons ce type de modélisation pour nos deux premières méthodes de génération. Le modèle que nous utilisons pour ces méthodes de génération est GPT-2 [2] (Radford *et al.*, 2019). Il existe d'autres modèles entraînés sur d'autres tâches. Dans notre cas, nous utiliserons aussi T5 (Raffel *et al.*, 2020), qui a été entraîné en utilisant un objectif de remplacement de séquences. Nos deux dernières méthodes de génération utilisent cette capacité de remplacement pour éditer la mention ou le contexte d'une phrase étiquetée ou pseudo-étiquetée. Pour ces méthodes, nous utilisons T5 v1.1 [3]. Toutes les méthodes de génération sont appliquées à des phrases étiquetées ou pseudo-étiquetées. Chaque méthode est illustrée par un exemple pour lequel le rouge indique les portions provenant directement de la phrase d'origine, le bleu indique les portions générées, et le gras indique les portions annotées.

génération de phrase suivante : Pour la génération de phrase suivante, nous utilisons le texte d'une phrase étiquetée ou pseudo-étiquetée comme contexte gauche pour modéliser le texte suivant. Notre hypothèse est que les nouvelles phrases apportées à l'ensemble de données par cette méthode apporteront de la variété à l'ensemble tout en restant dans le domaine.

Ceci est un **exemple**. → Il permet de décrire la situation.

Complétion de phrase : La complétion est une méthode utilisant la modélisation de gauche à droite pour compléter la phrase. La position à partir de laquelle générer est choisie par échantillonnage entre les premiers tokens des mentions dans la phrase. Si la phrase ne contient aucune mention, la position est échantillonnée entre 25% et 75% de la longueur de la phrase. Nous supposons que cette méthode apportera de la variété aux mentions tout en conservant l'approche classique de modélisation.

Ceci est un **exemple**. → Ceci est un objet d'intérêt.

Remplacement de mentions : Notre objectif est d'apporter la diversité aux objets d'intérêt en remplaçant les mentions des phrases étiquetées. Si plus d'une mention est présente, nous échantillonnons la mention qui doit être remplacée. Nous ignorons les phrases sans mention. Notre hypothèse est que les modèles de langue ont appris des informations sur n'importe quel sujet pendant leur entraînement. Les sujets qui nous intéressent sont liés aux concepts que nous souhaitons étiqueter dans les phrases. Nous supposons qu'étant donné un contexte qui contient une instance d'un concept donné, le modèle sera capable de remplacer cette instance par d'autres instances du même concept. Nous pensons que cette méthode fonctionnera mieux en combinaison avec d'autres méthodes.

Cet **exemple** est plus adapté. → Cet objet d'intérêt est plus adapté.

Remplacement de contexte : Le remplacement du contexte fonctionne de manière similaire au remplacement des mentions, sauf que c'est le texte non étiqueté qui est remplacé. Nous créons une liste de séquences contenant toutes les séquences de contexte de la phrase. Nous échantillonnons la séquence qui sera remplacée. Nous ignorons les phrases sans mention. Notre hypothèse est que les

2. https://huggingface.co/gpt2-large
3. https://huggingface.co/google/t5-v1_1-large

modèles d'étiquetage apprennent à prédire à la fois à partir du contexte et des mentions. De cette hypothèse, nous déduisons que l'apprentissage à partir de contextes plus variés pourrait être bénéfique pour l'étiqueteur.

Ceci est un **exemple**. → Nous avons besoin d'un exemple.

5 Ressources

	nb. phrases train	nb. phrases test	BERT large	BioBERT	SOTA
CoNLL	14 986	3 683	**92,0**		94,6
I2B2	11 482	27 625	85,0	**86,6**	90,3

TABLE 1 – Performances des modèles utilisés en score F_1 quand ils sont entraînés sur le corpus naturel complet et comparaison avec l'état de l'art. Métrique calculée avec seqeval (Ramshaw & Marcus, 1995; Nakayama, 2018). État de l'art pour CoNLL (Wang *et al.*, 2021a) et I2B2 (Si *et al.*, 2019) d'après papers with code [5].

Données Nous testons notre méthode sur deux corpus, CoNLL 2003 English (Sang & De Meulder, 2003) et I2B2 (Uzuner *et al.*, 2011). CoNLL est un corpus de nouvelles Reuters en anglais dans lequel des personnes, des lieux, des organisations et diverses entités sont étiquetées. I2B2 est un corpus de dossiers médicaux en anglais dans lequel les traitements, les problèmes et les tests sont étiquetés. Notre travail vise à permettre l'entraînement de modèles à faibles ressources. Nous testons notre méthode sur des corpus disponibles afin d'obtenir des résultats pouvant être comparés aux méthodes de l'état de l'art. Pour simuler un environnement à faibles ressources, nous devons restreindre la quantité de données disponibles pour l'entraînement. Nous échantillonnons les données pour obtenir des ensembles d'entraînement réduits. Nous répétons nos expériences sur un ensemble de graines afin de prendre en compte les biais d'échantillonnage dans l'évaluation de la méthode.

Modèles L'architecture des modèles que nous utilisons est BERT avec un classificateur MLP, ce qui nous permet de bénéficier de la vaste gamme de modèles BERT pré-entraînés. Nous affinons les modèles sur les données d'apprentissage, que ce soit pour la baseline ou les méthodes d'augmentation testées avec les paramètres présenté dans BERT (Devlin *et al.*, 2018). Nous avons testé quelques modèles BERT pré-entraînés sur les corpus complets afin de sélectionner les modèles que nous utiliserons (voir Tableau 1). Nous avons opté pour BERT Large (Devlin *et al.*, 2018) pour CoNLL, et BioBERT (Lee *et al.*, 2020) pour I2B2.

6 Expériences

Nous avons conçu un ensemble d'expériences pour évaluer le tri-apprentissage génératif. Ces expériences comparent l'apprentissage supervisé au tri-apprentissage génératif et avec les données

5. https://paperswithcode.com/

de tâches disponibles. La génération avec semi-supervision est testée avec deux algorithmes d'apprentissage semi-supervisé : le tri-apprentissage et l'auto-apprentissage. Notre baseline est l'apprentissage supervisé sur des données étiquetées sans aucune semi-supervision. Notre **topline est le tri-apprentissage** avec des données naturelles comme données non étiquetées. Pour la topline, nous considérons un ensemble de 10 000 phrases de chaque corpus à partir duquel nous extrayons les corpus supervisés S_n, composé de 50 à 1 000 phrases, et le reste est utilisé comme l'ensemble non étiqueté servant à peupler les L_i^t. Dans ces cas plus ou moins extrêmes, la quantité de données n'est pas suffisante pour apprendre des modèles performants, comme le montrent les résultats de la baseline Table 2.

À chaque épisode d'apprentissage semi-supervisé, nous générons une quantité fixe de données. Nous visons à générer 5 000 exemples pour chaque épisode. Cette quantité est suffisamment importante pour apporter une quantité significative d'informations mais pas trop afin d'éviter de ralentir le processus d'apprentissage. Nous générons une plus petite quantité de données pour les premiers épisodes car la quantité de contexte disponible, les données étiquetées et pseudo-étiquetées, est plus petite. Cette petite quantité de données pourrait également voir son signal noyé si trop de données sont générées, comme on le voit avec la méthode topline. Nous facilitons les deux premières étapes en générant seulement 500 et 2 500 exemples au lieu de 5 000.

La génération se fait dans le contexte de l'apprentissage semi-supervisé. Pour le tri-apprentissage, les exemples générés sont répartis en trois ensembles, un pour chaque modèle entraîné dans le tri-apprentissage. Les nouveaux ensembles de données sont générés en appliquant la méthode de génération respective aux exemples échantillonnés de l'ensemble de données naturelles et aux ensembles pseudo-étiquetés de l'épisode précédent. Les trois modèles étiquettent ensuite le texte généré par la méthode de génération. Si deux modèles ou plus parviennent à un accord, les nouvelles données sont conservées avec les pseudo-étiquettes. Ces données pseudo-étiquetées sont envoyées soit au paquet du modèle qui n'est pas d'accord si deux modèles sont d'accord, soit à l'un des ensembles au hasard si les trois modèles sont d'accord. Si aucun accord n'est trouvé, les données sont rejetées. Nous continuons à générer de nouveaux exemples jusqu'à ce que nous atteignions la quantité attendue ou un nombre maximal d'opérations de génération autorisé que nous avons fixé à 10 000 pour éviter de générer indéfiniment. En pratique, cette limite n'est jamais atteinte. Chaque méthode de génération génère cinq nouvelles phrases pour chaque phrase d'entrée.

7 Résultats

Les résultats de la baseline et les Δ entre la méthode présentée et la baseline sont présentés dans le tableau 2. Par souci de place, nous n'avons pas indiqué les résultats obtenus sur des sous-ensembles de taille 1000, mais ils conservent les tendances observés. Les résultats de la topline sont bien supérieurs aux résultats du supervisé. Les résultats du supervisé vont de 60,0 F_1 avec des sous-ensembles de taille 50 à 84,6 F_1 avec une taille de 500 pour CoNLL avec des gains décroissant de 8,6 à 3,8 pour la topline. Pour I2B2, les résultats de la baseline sont plus petits que ceux de CoNLL, allant de 39,3 à 77,6 F_1 score avec des gains. Les gains de I2B2 vont de 8,9 à la taille 100 à 5,4 à la taille 500 et sont légèrement inférieurs pour 50 phrases, avec 8,3.

Les résultats décrits dans cette section se trouvent dans le Tableau 2 pour les ressources faibles, et dans le Tableau 4 pour les corpus complets. Pour le tri-apprentissage, chaque méthode affiche des gains positifs à chaque taille de sous-ensemble et certaines dépassent même la topline. C'est le cas

Crp.	CoNLL				I2B2			
Taille	50	100	250	500	50	100	250	500
bsl.	60,0 2,9	70,5 4,7	81,3 1,7	84,6 1,2	39,3 3,1	50,4 1,4	63,8 0,7	71,5 1,0
Δtl	+8,6 2,5	+5,6 2,0	+4,2 0,7	+3,8 1,0	+8,3 2,2	+8,9 2,2	+6,7 1,2	+5,4 0,7
	Tri-apprentissage							
Δfl	+8,9 2,6	+4,4 2,8	+3,1 0,4	+2,8 1,0	+6,3 2,6	+4,5 2,6	+4,2 1,6	+3,6 1,0
Δcp	+7,8 1,9	+4,7 4,0	+2,7 1,0	+2,9 0,7	+6,3 1,6	+5,6 2,8	+4,9 1,5	+4,3 1,7
Δm	+9,0 2,1	+4,1 2,6	+2,8 1,1	+2,6 0,8	+10,4 4,5	+7,4 2,9	+7,1 1,7	+4,6 1,0
Δcr	+7,3 1,9	+2,9 2,3	+1,2 0,4	+1,7 0,7	+5,6 1,5	+7,7 3,1	+6,0 1,2	+4,2 1,0
Δcb	+9,3 2,7	+4,6 2,6	+3,4 1,0	+2,9 1,0	+7,7 1,5	+6,4 1,7	+5,7 1,9	+4,4 1,2
	Auto-apprentissage							
Δfl	+4,8 2,9	+1,4 3,6	-0,1 1,1	+0,6 1,0	+0,8 1,7	+0,2 2,3	-0,7 1,3	-0,2 0,5
Δcp	+3,9 2,7	+0,8 2,0	-0,4 1,0	+0,6 0,9	-0,5 1,6	+0,3 2,9	-1,4 0,8	-0,9 1,8
Δm	+6,0 2,9	+2,5 2,8	+0,7 0,6	+0,8 0,4	+4,5 2,6	+2,9 1,8	+3,0 1,1	+2,0 1,1
Δcr	+4,3 2,0	-0,5 4,0	-1,1 1,2	-0,2 0,7	+3,6 1,8	+3,6 2,5	+3,9 0,9	+2,2 0,7
Δcb	+4,9 3,5	+1,4 3,1	-0,4 0,4	+0,9 1,2	+1,6 1,0	+0,7 1,7	-0,1 1,2	+0,6 1,6

TABLE 2 – Δ de F_1 des différentes méthodes comparées à l'apprentissage supervisé. Les écarts type sur les cinq seeds utilisées sont rapportés en plus petit. Bsl. et tl sont les résultats obtenus avec la baseline et la topline. Les abréviations fl, cp, m, cr, et cb décrivent la génération de phrase suivante, la complétion, le remplacement de mentions, le remplacement de contexte, et l'expérience combinée.

des méthodes de génération de phrase suivante et de remplacement de mention pour CoNLL avec des gains moyens de 8,9 et 9,0 F_1. C'est également le cas pour la méthode de remplacement de mention pour I2B2 avec des gains moyens de 10,4 et 7,1 F_1 score aux tailles 50 et 250, respectivement. Les gains suivent les mêmes tendances que la topline, avec des rendements décroissants plus la taille du sous-ensemble est élevée. Pour CoNLL, le remplacement de contexte est la méthode de génération la moins performante dans l'ensemble. Pour I2B2, avec un sous-ensemble d'apprentissage de taille 50, le remplacement de contexte est la méthode la moins performante. Néanmoins, avec des quantités de données plus importantes, elle devient la deuxième méthode la plus performante, et la génération de phrase suivante devient la moins performante. Le remplacement de mentions est la meilleure méthode de génération individuelle sur I2B2, avec les meilleurs ou les seconds meilleurs résultats pour chaque taille de sous-ensemble. Dans l'ensemble, les méthodes fondées sur GPT-2 donnent des résultats inférieurs aux méthodes fondées sur T5v1.1 sur I2B2.

Les gains avec l'auto-apprentissage sont inférieurs à leurs homologues du tri-apprentissage, avec quelques résultats négatifs. La génération de mentions reste la meilleure méthode individuelle, car c'est la meilleure méthode pour CoNLL et elle est en concurrence avec le remplacement de contexte en fonction de la taille du sous-ensemble pour I2B2.

La méthode combinée est réalisée en échantillonnant la méthode de génération parmi les méthodes disponibles chaque fois que nous voulons générer. La méthode présente deux comportements différents sur les deux corpus. Pour CoNLL, la méthode de génération combinée est la meilleure méthode de génération. Pour I2B2, la méthode de génération combinée s'apparente davantage à une méthode

	Suivante	Complétion	Mentions	Contexte	Combiné
Suivante		0.68	0.96	0.81	0.92
Complétion	**0.35**		0.79	0.63	0.69
Mention	**0.05**	**0.22**		**0.20**	**0.26**
Contexte	**0.21**	**0.38**	0.82		0.51
Combiné	**0.11**	**0.33**	0.76	0.51	

TABLE 3 – Score de l'Almost Stochastic Dominance (Dror *et al.*, 2019) du tri-apprentissage à faible ressources avec indice de significativité de 0.05. Nous testons si la méthode de la ligne A est dominante sur la méthode de la colonne B. A est dominante sur B si le score est inférieur à 0.5.

	baseline	complétion	suivante	mention	contexte	combiné
CoNLL	92,0	92,1	91,8	92,2	92,2	**92,3**
I2B2	86,6	87,5	**88,0**	87,7	87,6	87,5

TABLE 4 – F_1 scores des méthodes de génération sur les corpus complets. Une seed a été utilisée.

moyenne. Elle est par ailleurs inférieure ou égale aux méthodes de remplacement, à l'exception du remplacement de contexte sur le plus petit sous-ensemble. La génération combinée est supérieure aux méthodes de modélisation de gauche à droite pour I2B2. Dans l'ensemble, la méthode combinée a un facteur de corrélation de Pearson plus élevé pour les méthodes de suivi et de complétion avec 0,98 et 0,97 que pour la mention et le remplacement de contexte avec 0,92 et 0,80, respectivement. Ces coefficients ont été calculés avec les résultats rapportés dans le Tableau 2 et les résultats sur les sous-ensembles de taille 1000. Nous avons utilisé le test d'Almost Stochastic Dominance(ASD) (Dror *et al.*, 2019) pour déterminer quelle est la meilleure méthode de génération sur nos sous-ensembles de données avec un indice de significativité de 0.05. D'après les résultats montrés Table 3, le remplacement de mentions est dominant sur toutes les autres méthodes et est donc la méthode à privilégier. La complétion n'est dominante que sur la génération de phrase suivante, qui ne domine aucune méthode. Le remplacement de contexte et la méthode combiné sont équivalentes.

À but indicatif, nous avons calculé des résultats pour le tri-apprentissage génératif sur les corpus de taille complète Table 4. Pour CoNLL, nos meilleurs résultats sont avec la méthode combinée avec une amélioration de 0,3. Nous obtenons cependant une détérioration de 0,2 avec la génération de phrase suivante. Globalement, sur CoNLL grandeur nature, nous obtenons des résultats positifs mais non significatifs. Pour I2B2, toutes les méthodes augmentent significativement le score F_1 avec des améliorations allant de 0,9 pour la complétion et la méthode combinée à 1,4 pour le suivi.

8 Conclusion

Dans cet article, nous avons proposé et évalué une méthode permettant d'allier génération de données et semi-supervision dans le but de réduire la dépendance aux données pour l'apprentissage étiqueté. Nous avons montré que notre méthode fonctionnait bien avec le tri-apprentissage, et moins bien avec l'auto-apprentissage. Nous avons testé différentes méthodes de génération découlant de la modélisation de langue pour générer des données. Parmi ces méthodes, le remplacement de mention

par T5 est la méthode ayant montré les meilleurs résultats sur nos deux corpus de test et domine les autres méthodes présentées d'après le test ASD. Cependant, utiliser des données naturelles de la tâche comme données non-annotées produit presque toujours de meilleurs résultats. Il reste cependant beaucoup de questions auxquelles nous souhaiterions répondre dans de futurs travaux. Nous avons montré que la génération permettait de transformer une tâche supervisée en tâche semi-supervisée avec des gains de performances. Il peut être intéressant d'analyser l'impact des données produites de cette manière dans un cadre supervisé.

Références

BLUM A. & MITCHELL T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, p. 92–100.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

DING B., LIU L., BING L., KRUENGKRAI C., NGUYEN T. H., JOTY S., SI L. & MIAO C. (2020). Daga : Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv :2011.01549*.

DOPIERRE T., GRAVIER C. & LOGERAIS W. (2021). Protaugment : Intent detection meta-learning through unsupervised diverse paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 2454–2466.

DROR R., SHLOMOV S. & REICHART R. (2019). Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)* : Association for Computational Linguistics.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). Biobert : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240.

NAKAYAMA H. (2018). seqeval : A python framework for sequence labeling evaluation. Software available from https ://github.com/chakki-works/seqeval.

NEURAZ A., LLANOS L. C., BURGUN A. & ROSSET S. (2018). Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. *CoRR*, **abs/1811.09417**.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, **1**(8), 9.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, **21**, 1–67.

RAMSHAW L. & MARCUS M. (1995). Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

RUDER S. (2019). *Neural Transfer Learning for Natural Language Processing*. Thèse de doctorat, National University of Ireland, Galway.

RUDER S. & PLANK B. (2018). Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1044–1054.

SANG E. T. K. & DE MEULDER F. (2003). Introduction to the conll-2003 shared task : Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, p. 142–147.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 86–96.

SI Y., WANG J., XU H. & ROBERTS K. (2019). Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, **26**(11), 1297–1304.

SØGAARD A. (2010). Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, p. 205–208.

UZUNER Ö., SOUTH B. R., SHEN S. & DUVALL S. L. (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA*, **18**(5), 552.

VAN ENGELEN J. E. & HOOS H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, **109**(2), 373–440.

WANG X., JIANG Y., BACH N., WANG T., HUANG Z., HUANG F. & TU K. (2021a). Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 2643–2660.

WANG Y., MUKHERJEE S., CHU H., TU Y., WU M., GAO J. & AWADALLAH A. H. (2021b). Meta self-training for few-shot neural sequence labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, p. 1737–1747.

WEI J. & ZOU K. (2019). EDA : Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 6382–6388, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1670.

YAROWSKY D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, p. 189–196.

ZHOU Z.-H. & LI M. (2005). Tri-training : Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, **17**(11), 1529–1541.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Évaluation d'un générateur automatique de reformulations médicales

Ioana Buhnila[1] Amalia Todirascu[1]
(1) LiLPa UR 1339, Université de Strasbourg, 67000 Strasbourg, France
ioana.buhnila@etu.unistra.fr, todiras@unistra.fr

RESUME

Les textes médicaux sont difficiles à comprendre pour le grand public à cause des termes de spécialité. Ces notions médicales ont besoin d'être *reformulées* en utilisant des mots de la langue commune. La **reformulation** représente le processus de réécriture qui a le rôle d'expliquer ou simplifier une phrase ou syntagme. Nous présentons la méthodologie de construction d'un jeu de données original (termes et reformulations) permettant la détection et génération des nouvelles reformulations médicales. Pour compléter ce corpus, nous menons des expériences de génération automatique de reformulations médicales sous-phrastiques avec l'outil **APT** (Nighojkar & Licato, 2021), qui s'appuie sur des techniques d'apprentissage profond. Nous adaptons le modèle de langue de type Transformer **T5** (Raffel et al., 2020) avec des termes médicaux et leurs reformulations annotés manuellement en français et en roumain, langue romane peu dotée en ressources pour le TAL. Nous présentons une analyse détaillée des résultats de la génération automatique des paraphrases.

ABSTRACT

Evaluation of an automatic medical paraphrase generator
Medical texts are difficult to understand for laypeople due to technical terms. Medical concepts need to be *paraphrased* using words from the common language. **Paraphrasing** is the process of rewriting with the aim of explaining or simplifying a sentence or phrase. We present the method for manually building a dataset for paraphrase detection and generation. We are conducting experiments on automatic generation of sub-phrastic medical paraphrases with the machine learning tool **APT** (Nighojkar & Licato, 2021), using deep learning techniques. We train the Transformer **T5** language model (Raffel et al., 2020) with manually annotated medical terms and paraphrases in French and Romanian, a Romance language lacking resources for NLP applications. We present a detailed analysis of the results of the paraphrase generation.

MOTS-CLES : génération automatique de la reformulation, apprentissage automatique, texte médical, Transformer T5
KEYWORDS: paraphrase generation, machine learning, medical text, T5 Transformer

1 Introduction

Les connaissances médicales de spécialité sont facilement consultables en ligne. Pourtant, elles ne sont pas toujours facilement *compréhensibles* pour le grand public. Cette difficulté est due au grand nombre de *termes médicaux* employés dans ces textes. *Le terme* est une unité lexicale de spécialité

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 80–93.

qui représente des connaissances spécifiques à un domaine du savoir (Costa, 2005). La terminologie médicale est difficile à comprendre pour un non-spécialiste à cause de l'origine grecque ou latine de dénominations, par exemple le terme « myocardiaque », formé avec une base latine « myo » (muscle) et une base grecque « cardia » (cœur) (Grabar & Hamon, 2016). Ces connaissances spécialisées sont faciles à comprendre pour un spécialiste, mais leur sens reste souvent obscur pour le grand public, d'où la nécessité d'une vulgarisation constante dans ce domaine (Grabar & Hamon, 2016). *Vulgariser* la médecine représente le processus de simplification lexicale et syntaxique qui a comme but de rendre les notions médicales très techniques compréhensibles en fonction du public cible. Informer correctement le grand public sur les questions médicales est une tâche qui demande un effort soutenu, vu l'innovation continue et les défis sanitaires constants (Pecout et al., 2019).

Nous considérons que les reformulations sont nécessaires dans la vulgarisation scientifique (Vargas, 2008 ; Cardon, 2018). *La reformulation* est un processus linguistique de transformation du discours qui a le rôle d'expliquer, simplifier ou pointer une phrase ou un syntagme. Nous travaillons sur des *reformulations sous-phrastiques*, ne dépassant pas la portée de la phrase, qui se présentent sous forme d'explications, de paraphrases (syntagmes synonymiques) ou de définitions de termes scientifiques monolexicaux et polylexicaux, de type « <terme>placebo</terme> ou <ref>absence d'intervention</ref> ». Notre objectif est double : de *construire* manuellement *un corpus valide* de *termes médicaux et leurs reformulations* (au sens large du terme) et de *générer automatiquement d'autres reformulations médicales* à partir de nos données annotées précédemment sur deux corpus, en français et en roumain, et ainsi agrandir automatiquement nos corpus de reformulations. Ce qui nous motive davantage dans notre recherche est le nombre restreint de ressources pour la reformulation dans le domaine médical pour la simplification de textes (Cardon, 2018 ; Cardon & Grabar, 2019), particulièrement pour des langues de moindre diffusion comme le roumain (Buhnila, 2021).

Nous présentons dans la section 2 l'état de l'art sur la génération des reformulations, suivi par notre méthodologie et les données exploitées (dans la section 3). Dans la section 4 nous présentons nos expériences de génération automatique, l'analyse et l'évaluation des résultats de génération de reformulations médicales. La conclusion, la comparaison avec l'état de l'art et des perspectives de recherches sont présentées dans la section 5.

2 Contexte

Certaines études ont été réalisées sur la reformulation dans le domaine de la médecine (Elhadad & Sutaria, 2007 ; Deléger & Zweigenbaum, 2009 ; Grabar & Hamon, 2015). Des ressources lexicales comme WordNet (Miller, 1998) pour la langue générale et UMLS (Bodenreider, 2004) ou Snomed (Spackman et al., 1997 ; Donnelly, 2006) pour le domaine médical sont utiles pour l'identification automatique des paraphrases selon leur degré de synonymie (Cardon & Grabar, 2019 ; Koptient et al., 2019). Il existe plusieurs travaux sur la simplification lexicale automatique des textes écrits (Specia et al., 2012 ; Shardlow, 2014 ; Grabar & Hamon, 2015 ; Saggion, 2017 ; Cardon, 2021). Ces études sont réalisées sur des termes simples et majoritairement sur la paraphrase phrastique. Nous travaillons sur les reformulations sous-phrastiques de termes médicaux simples et polylexicaux. Nous prenons en compte des reformulations variées (définitions, exemples, explications) qui sont très différentes au niveau lexical et syntaxique par rapport à l'original. Notre but est de construire un corpus de reformulations qui illustre ces phénomènes et de générer des nouvelles reformulations médicales.

Dans ce sens, nous nous intéressons à l'apprentissage automatique par réseaux de neurones, car plusieurs méthodes peuvent être appliquées pour travailler sur la paraphrase :

- *Similarité sémantique* textuelle (STS), qui mesure le degré d'équivalence des textes ou des phrases similaires qui contiennent des mots en commun (Agirre et al., 2016) ;
- *Identification de paraphrase (PI)*, qui identifie si deux phrases ou segments ont le même sens (Brockett & Dolan, 2005 ; Xu et al., 2015) ;
- *Génération des paraphrases (PG)*, qui crée de nouveaux textes à partir des données d'entrée et des modèles de langues (Gupta et al., 2018 ; Bowman et al., 2016).

Les méthodes basées sur des mesures de similarité textuelle (STS ou PI) comptabilisent les mots qui ont une forme similaire. Ces méthodes ont des limites quant à l'identification des paraphrases qui utilisent des mots ou des phrases avec des formes très différentes. Ainsi, nous adoptons une architecture neuronale de type paradigme contradictoire, qui vise à identifier des *différences lexicales et syntaxiques*, qui permet aussi de *générer des paraphrases* (PG) pour créer des reformulations les plus diverses possibles, en gardant le plus possible le même contenu sémantique (moins pour les reformulations de type explication). Nous considérons cette méthode la plus adaptée à notre tâche d'identification de la reformulation, vu la grande diversité des reformulations médicales identifiées dans nos corpus : paraphrases, exemplifications, synonymes, de type définition, explication ou abréviation. Nous présentons en détail cette architecture dans le point suivant.

2.1 L'architecture neuronale APT

Pour générer des reformulations à partir du terme médical, nous faisons appel à l'architecture neuronale **APT** (*Adversarial Paraphrasing Task*) (Nighojkar & Licato, 2021). Cette architecture utilise une méthode pour générer des reformulations composées avec des significations équivalentes mais présentant des différences lexicales et syntaxiques. Ce modèle vise à identifier le sens général d'une phrase, non pas uniquement le sens des mots séparés. L'architecture **APT** est construite sur deux principes :

- *La similarité de sens* : cette similarité se vérifie par le fait que deux phrases qui sont mutuellement implicites sont sémantiquement équivalentes, et sont donc, des paraphrases ;
- *La dissimilarité de la structure* : mesurée avec BLEURT (Sellam et al., 2020), un score qui évalue les textes générés automatiquement en se basant sur les plongements des mots du modèle de langue BERT (Devlin et al., 2019). Ce score évalue la similarité lexicale et syntaxique de chaque paire de phrases. BERT est adapté pour la tâche d'évaluation de la qualité de la prédiction automatique par rapport à l'évaluation humaine. Le score BLEURT utilise un ensemble réduit de données annotées par les humains pour adapter BERT à cette tâche d'évaluation. Pourtant, les données de référence utilisées pour créer BLEURT ont été générées automatiquement par plusieurs types de transformations : mot ou segment de phrase masqué et remplacé à l'aide de BERT, traduction et rétrotraduction de la phrase et suppression aléatoire des mots.

Dans leur étude, Nighojkar et Licato (2021) mènent les expériences sur des paraphrases phrastiques en anglais, appartenant au langage général. Ils utilisent un score MI (implication mutuelle) entre 2 segments, qui mesure les inférences nécessaires pour déduire le sens du premier à partir du deuxième et vice-versa, à l'aide du modèle de langue **T5-base**. Ainsi, ils sélectionnent des paraphrases qui ont un score d'implication mutuelle MI grand et un score BLEURT (score de similarité de phrases) réduit. Les expériences avec les corpus anglais annotés manuellement ont donné des meilleurs résultats par rapport aux données extraits automatiquement de Twitter. Nous testons ce modèle contradictoire d'identification de la reformulation (**APT**) sur nos données sous-phrastiques (terme médical versus reformulation), en français et en roumain, appartenant au domaine médical. Par rapport à Nighojkar et Licato (2021), nous travaillons avec des reformulations sous-phrastiques présentant peu de similarités lexicales ou syntaxiques avec l'original.

Pour lancer nos expériences de génération, nous adoptons des modèles de langues de type Transformer, en faisant appel aux données annotées manuellement, plus fiables que des jeux de données de grande taille acquises automatiquement. Nous présentons ce concept et notre méthode par la suite.

2.2 Modèle de langues de type Transformer

Un *modèle de langue* est une représentation des informations linguistiques variées qui participent à la construction du sens dans une langue donnée par des règles linguistiques et des normes d'usage spécifiques à cette langue. Ces normes sont converties en vecteurs (des valeurs numériques attribuées aux mots et séquences de mots) pour les rendre exploitables par une machine. Le modèle de langue estime la probabilité qu'un mot ou un syntagme soit présent dans une langue, en fonction du contexte. Les *Transformers* (Vaswani et al., 2017) sont des modèles de langues pour l'apprentissage automatique profond par des réseaux de neurones. Les *Transformers* traitent les données de manière séquentielle (position des mots) à l'aide de mécanismes d'auto-attention et de systèmes de type encodeur-décodeur (chaque mot reçoit un vecteur numérique).

Nous utilisons le modèle de langue **T5** (*Text-to-Text Transformer*) de Google (Raffel et al., 2020), car il contient quatre langues, dont nos langues d'étude : l'allemand, l'anglais, le français et le roumain. Ce Transformer est pré-entrainé sur C4 (*Colossal Clean Crawled Corpus)* un corpus nettoyé avec une taille colossale de 7 téraoctets, extrait du corpus Web de Common Crawl. **T5** a été pré-entrainé pour plusieurs tâches spécifiques du TAL, dont l'identification de la paraphrase et la similarité de phrases. Nous l'adaptons pour notre propre jeu de données (nous travaillons sur des reformulations sous-phrastiques au lieu des reformulations phrastiques) dans un domaine spécifique. Nous présentons notre méthode et les données exploitées pour générer des reformulations médicales dans le point suivant.

3 Méthodologie

Afin de générer des reformulations spécifiques au domaine médical, nous devons adapter le modèle de langue **T5** à l'aide de données médicales. Nos données sont des paires (terme médical, reformulation médicale), en français et en roumain, données que nous avons extrait semi-automatiquement à l'aide des ontologies et des reformulations. Ces données sont annotées manuellement, suivant la méthodologie proposée par (Buhnila, 2021 ; 2022b). Nous adaptons le modèle **T5** pour le français, ainsi que pour le roumain et nous utilisons l'architecture **APT** (Nighojkar & Licato, 2021) pour la génération de prédictions de reformulations médicales. Ces prédictions sont analysées et évaluées en suivant notre échelle de lisibilité pour élargir les corpus de reformulations. Nous détaillons ces étapes et les données utilisées par la suite.

3.1 Données médicales annotées

Pour le français nous travaillons sur deux corpus : **CLEAR Cochrane** (Grabar & Cardon, 2018) et **ClassYN** (Todirascu et al., 2012). Ces corpus écrits et comparables sont constitués des textes scientifiques du domaine médical destinés aux experts et des textes simplifiés pour le grand public. Le corpus **CLEAR Cochrane** a une taille de **4 355 054** tokens et le corpus **ClassYN** de **1 779 423** tokens. Pour la langue roumaine, nous exploitons le corpus **GrandMed-Ro** (Buhnila, 2018) constitué à partir des textes de vulgarisation extraites de la toile avec Sketch Engine (Kilgarriff et al., 2014) (6 440 951 tokens).

Nous identifions automatiquement les termes médicaux dans notre corpus français avec l'annotateur **SIFR-BioPortal** (Tchechmedjiev et al., 2018), utilisant l'ontologie médicale **SNOMED-3.5VF** (Côté, 1998) (qui contient 150 906 concepts médicaux) et des scripts en Perl pour l'extraction des phrases contenant les termes. À notre connaissance, il n'existe pas de liste de termes médicaux en roumain en libre accès. Pour cela, nous avons extrait les entités nommées médicales du corpus médical annoté **MoNERo** (Mitrofan et al., 2019) et nous avons créé une **terminologie médicale en roumain** de 14 133 termes. Notre méthode consiste dans l'extraction automatique de phrases qui contiennent simultanément des termes médicaux et des *marqueurs lexicaux ou grammaticaux de reformulation*, de type « c'est-à-dire », « autrement dit », « encore appelé », « est un », « également appelé », « désigne », signifie », etc., et leurs équivalents roumains (Grabar & Hamon, 2015 ; Antoine & Grabar, 2016 ; Buhnila, 2022a). Nous prenons en compte également les *marqueurs orthographiques* comme les doubles points ou les parenthèses.

Les phrases qui contiennent des termes médicaux et des marqueurs de reformulations sont extraites automatiquement. Ces phrases sont par la suite annotées semi-automatiquement et analysées manuellement du point de vue lexical et sémantico-pragmatique afin d'identifier des reformulations de termes médicaux (selon les critères de Buhnila, 2021 ; 2022a ; 2022b). Nous définissons les *relations lexicales* comme le lien lexical qui existe entre les deux segments, le terme médical et la reformulation. Nous analysons et annotons les relations lexicales de type *synonymie, hypéronymie, hyponymie et méronymie*[1], dans le contexte du texte médical (Condamines, 2018 ; Ramadier, 2016 ; Săpoiu, 2013). *Les fonctions sémantico-pragmatiques* représentent les raisons qui poussent le locuteur à utiliser la reformulation dans les textes écrits du domaine médical (de type *définition, paraphrase, synonymie, dénomination*[2]*, exemplification, explication*) (Malaise et al., 2004 ; Eshkol-Taravella & Grabar, 2017 ; Buhnila, 2022a). Les phrases ont été annotées par au moins deux annotateurs non-spécialistes du domaine de la médecine. L'accord inter-annotateur Kappa (Cohen, 1960) pour l'annotation des phrases contenant des reformulations valides ou sans reformulation est, pour le français, de **0,78** (**0,62** pour toutes les phrases du corpus CLEAR et **0,95** pour ClassYN), et pour le roumain, de **0,82** (pour 1 010 phrases annotées du corpus GrandMed-Ro).

Afin de construire des listes de reformulations correctes pour chaque langue, nous avons réalisé une adjudication entre les deux annotations. Lorsqu'il y avait des différences dans les deux annotations (statut *oui* versus statut *non*), nous avons choisi le statut de plus adapté de la reformulation en suivant le guide d'annotation. À l'issue de nos analyses, nous avons constitué une liste de **8 626** paires de *terme – reformulation* extraites des corpus français CLEAR Cochrane et ClassYN et **3 027** paires du corpus roumain GrandMed-Ro.

3.2 Données d'entraînement

Nous avons lancé nos expériences avec la plateforme **APT** et la version **T5-base** du Transformer qui a 220 millions de paramètres, permettant des prédictions de bonne qualité en anglais. Nous l'adaptons pour le français et le roumain en s'appuyant sur les données annotées et validées manuellement. Pour chaque expérience monolingue, nous avons utilisé trois types de données :

1. *Un corpus d'entraînement* : **8 146** paires *terme – reformulation* pour le français et **2 727** paires pour le roumain ;

[1] La relation d'holonymie n'est pas présente dans les reformulations identifiées. Nous n'avons pas traité la relation d'antonymie, car nous cherchons des reformulations qui gardent une équivalence de sens et non pas des contraires.

[2] Notre définition de la *dénomination* : le terme est reformulé à l'aide d'un autre nom (ou terme), en gardant une relation lexicale d'équivalence sémantique (la synonymie), mais sans l'intention d'expliciter ou simplifier le terme reformulé.

2. *Un corpus de validation* : l'évaluation est réalisée pendant le processus d'apprentissage sur des blocs de vingt exemples corrects de paires *termes – reformulation* ;
3. *Un corpus de test* : **480** paires *terme – reformulation* extraites aléatoirement de la liste de reformulations du corpus français et **300** paires *terme – reformulation* du corpus roumain (car nombre réduit de phrases annotées). Ces exemples n'ont pas été utilisés pour l'entraînement pour éviter les biais.

Dans la section suivante nous présentons en détail les résultats de ces expériences, la précision des prédictions correctes pour les deux langues, le français et le roumain.

4 Expériences et résultats

Nous avons modifié les paramètres de l'architecture **APT** pour avoir entre 1 et 5 prédictions de reformulations médicales pour chacun de 480, respectivement 300 termes médicaux de la liste de test. L'adaptation du modèle **T5-base** avec les reformulations a duré 24 heures pour chacune de langues de notre étude. Nous avons utilisé une taille maximale de 256 mots (pour les reformulations), le taux d'apprentissage (3e-4), 4 epochs et des batchs (entrainement et validation) de taille de 20 paires, le paramètre concernant la réduction des poids (0,01), un optimiseur AdamW (l'epsilon 1e-8). Nous évaluons manuellement chaque prédiction et nous présentons les résultats de notre analyse.

4.1 Génération des reformulations médicales

Nous avons obtenu **2 268 prédictions** de reformulations médicales générées automatiquement pour les **480** termes médicaux du corpus de test en français, et **1 490 prédictions** en roumain pour les **300** termes de test. Par exemple, le terme médical « drépanocytose » est associé à deux types d'informations : la reformulation médicale issue de nos annotations manuelles « est une maladie héréditaire du sang causée par des anomalies de la production d'hémoglobine » (sous l'intitulé *Truth,* reformulation de référence) et des prédictions automatiques (*Prediction*) : « maladie du sang génétique » ; « , maladie héréditaire de l'hémoglobine », « d'autres maladies génétiques » ; « tel que maladie génétique » ; « : maladie d'Alzheimer ». Nous observons que pour cet exemple, les quatre premières prédictions sont correctes. Parmi les 480 termes médicaux en français, **180 (37,50%)** termes ont été attribués des prédictions de reformulations correctes. Pour le roumain, **80 (26,66%)** termes ont été bien reformulés par le modèle de langue **T5-base**. La faible précision pour le roumain s'explique par la taille réduite de données disponibles pour l'entraînement (2 727 reformulations annotées par rapport à 8 146 pour le français) et par la morphosyntaxe plus complexe de la langue (déclinaisons, cas marqués morphologiquement, signes diacritiques absents dans le corpus qui a été utilisé pour construire le modèle T5).

Prédiction avec T5-base	**CLEAR et ClassYN**		**GrandMed-Ro**	
	N° termes	**Précision**	**N° termes**	**Précision**
au moins une prédiction correcte	180	37,50%	80	26,66%
aucune prédiction correcte	300	62,50%	220	73,33%
Total	**480**	**100%**	**300**	**100%**

TABLE 1 : Résultats de génération automatique de prédictions avec T5-base

Afin d'évaluer la qualité des prédictions automatiques de reformulations, nous avons créé un **guide d'annotation** sur une **échelle d'annotation** permettant de les évaluer, que nous présentons en détail ci-dessous.

4.2 Evaluation des prédictions automatiques

L'échelle d'annotation conçue pour évaluer les prédictions est construite en trois parties :

1. *La première* évalue chaque prédiction de reformulation générée :
   - La valeur 2 : pour les prédictions identiques à la reformulation médicale initiale (Truth, qui est la valeur de référence) ;
   - La valeur 1 : pour les prédictions correctes, mais différentes de la reformulation médicale initiale ;
   - La valeur 0 : pour les prédictions incorrectes. Même s'il y a des parties de reformulations correctement construites, nous avons attribué la valeur 0 en cas de mots inventés ou des mots inadaptés dans le contexte (par exemple on parle de *l'inflammation de la peau* et on utilise des symptômes des *infections urinaires*) ;
   - La valeur -1 : pour les répétitions du terme médical.
2. *La deuxième* calcule la moyenne de toutes les prédictions d'un terme ;
3. *La troisième* évalue si au moins une des prédictions générées est correcte parmi toutes les reformulations générées pour un terme médical à la fois :
   - La valeur 1 : le terme médical a au moins une prédiction de reformulation médicale correcte ;
   - La valeur 0 : le terme médical n'a reçu que des prédictions de reformulations médicales qui ne sont pas correctes.

Nous présentons notre analyse et évaluation des résultats de prédictions et nous calculons la précision de génération automatique de reformulations du modèle T5-base.

4.3 Analyse de prédictions de reformulations

Nous analysons en détail les scores donnés à chacune de 2 268 prédictions générées automatiquement par le modèle T5-base (voir Table 2). Très peu de prédictions sont identiques à la valeur de référence (Truth) : pour 73 prédictions (3,21%) le modèle a généré correctement les reformulations correspondantes pour 27 termes médicaux. Ces prédictions sont de plusieurs types :

- **Abréviations** : Terme : *troubles associés à l'entorse cervicale* ; Truth: *(TAEC)* ; Prediction: *(TAEC)* ;
- **Hypéronymes** : Terme : *maladies cardiovasculaires, l'ostéoporose et la démence* ; Truth: *telles que maladies chroniques* ; Prediction: *maladies chroniques* ;
- **Paraphrases** : Terme : *syndrome confusionnel* ; Truth: *(délirium)* ; Prediction: *(délirium)* ;
- **Dénominations** : Terme : *sclérose latérale amyotrophique* ; Truth: */ maladie du motoneurone* ; Prediction: */ maladie du motoneurone*.

Dans les cas des *dénominations*, l'apparition fréquente de la paire respective de terme – reformulation dans le corpus d'apprentissage joue un rôle important dans la précision de la prédiction générée. Par exemple, nous comptons 16 prédictions générées automatiquement de type « maladie du motoneurone » pour le terme « sclérose latérale amyotrophique », qui apparait 11 fois dans la liste de termes du corpus de test et 41 fois ensemble avec le terme dans le corpus d'entraînement. Nous remarquons également que, même si APT a été paramétré pour générer maximum 5 prédictions pour chaque terme médical, il peut trouver la reformulation correcte exacte

(selon Truth) avec un seul essai, comme par exemple pour les termes médicaux polylexicaux « essais contrôlés randomisés » ; Truth: (ECR) ; Prediction: (ECR) ou « Organisation mondiale de la Santé » ; Truth: (OMS) ; Prediction: (OMS).

Échelle d'annotation des prédictions	**CLEAR et ClassYN**		**GrandMed-Ro**	
	N° prédictions	**Précision %**	**N° prédictions**	**Précision %**
Score 2	73	3,21%	3	0,2%
Score 1	244	10,75%	117	7,85%
Scores positifs	***317***	***13,97%***	***120***	***8,05%***
Score 0	1848	81,48%	1 320	88,59%
Score -1	94	4,14%	50	3,35%
Scores négatifs	***1942***	***85,62%***	***1 370***	***91,94%***
Total	**2 268**	**100%**	**1 490**	**100%**

TABLE 2 : Statistiques sur les scores de l'échelle d'évaluation de prédictions avec T5-base

Parmi toutes les prédictions générées pour le français, **317 (13,97%)** sont des reformulations correctes pour les **180 termes** ayant au moins une prédiction correcte (**37,50%**). **T5-base** a généré également d'autres reformulations correctes que celle de la colonne *Truth* parmi ses prédictions, **244 (10,75%)** plus précisément. Ainsi, nous avons obtenu des *nouvelles reformulations*, en dehors de celles qui ont été données comme exemples lors de l'entraînement. Concernant les prédictions correctes en roumain, nous observons que la majorité (**98,25%**) de prédictions générées sont des reformulations nouvelles (*score 1*) par rapport aux données d'apprentissage, les reformulations annotées (*score 2*). Par exemple, pour le terme « Infectiile herpetice » (les infections par l'herpès), dont la reformulation initiale était « sunt afectiuni eruptive de natura inflamatorie » (*sont des affections éruptives de nature inflammatoire*), **T5-base** a généré la reformulation « cum ar fi o infectie virala contagioasa » (*telle qu'une infection virale contagieuse*). Nous avons identifié 112 (7,51%) prédictions qui contiennent des répétitions de type « boala boala » *(maladie maladie)*, « un tip tip » *(un type type)*, « cum ar fi un virus viral viral » *(comme un virus viral viral)*, « fiind afectiune cronica cronica » *(étant une affection chronique chronique)*, « afectiune afectiune » *(affection, affection)*.

Pour éliminer les répétitions du même mot dans la prédiction générée, nous avons mené une deuxième expérience (**Exp 2**) avec **T5-base** et l'architecture **APT** (**sans répétition du même mot**). Nous avons évalué les prédictions générées lors de cette deuxième expérience pour le corpus français et le corpus roumain et nous comparons les deux versions (**Exp 1** et **Exp 2**, sans répétitions) ci-dessous. Pour le français, la précision des termes avec au moins une prédiction correcte a augmenté à **44,79%** (**7,29%** de plus que dans l'**Exp 1**). Pour le roumain, nous avons identifié **19% de termes supplémentaires avec au moins une prédiction correcte** pour obtenir une nouvelle précision de **45,66%** (Table 3).

Les prédictions correctes (*score 1* et *score 2*) restent quand même limitées : **381** (**16,80%**) pour le français et **222** (**14,94%**) pour le roumain. La contrainte sans répétitions imposée à la deuxième expérience augmente le nombre de prédictions correctes (*scores positifs*) de **2,83%** pour le français et de **6,89%** pour le roumain (voir Table 4).

	CLEAR et ClassYN				GrandMed-Ro			
Modèle	**T5-base Exp 1**		**T5-base Exp 2**		**T5-base Exp 1**		**T5-base Exp 2**	
Données statistiques	**N° trm**	**%**	**N° trm**	**%**	**N° trm**	**%**	**N° trm**	**%**
au moins une prédiction correcte	180	37,50%	**215**	**44,79%**	80	26,66%	**137**	**45,66%**
aucune prédiction correcte	300	62,50%	265	55,20%	220	73,33%	163	54,33%
Total	480	100%	480	100%	300	100%	300	100%

TABLE 3 : Statistiques sur les résultats de prédictions de l'expérience 1 (répétitions possibles) et 2 (sans répétitions)

Nous avons calculé le **score inter-annotateur Kappa** (Cohen, 1960) pour **1 196 prédictions** en français et **1 234 prédictions** en roumain (générées pour **250** termes de chaque langue), annotées par deux annotateurs francophones, non-spécialistes du domaine de la médecine. Nous avons obtenu un score inter-annotateur Kappa de **0,44** pour le français et de **0,48** pour le roumain. Ces accords sont *modérés* car ils concernent quatre valeurs d'annotation différentes selon le guide d'annotation de prédictions automatiques (*2, 1, 0* et *-1*). Par conséquent, ces scores sont très précis pour chaque valeur. Nous avons calculé également le score Kappa pour les **250 termes annotés par langue**. Pour les 250 termes annotées en français le score est de **0,42** et en roumain de **0,55**, également des scores inter-annotateur Kappa *modérés*. Les scores modérés s'expliquent par la difficulté de la tâche : il est difficile d'identifier la reformulation correcte (surtout quand les mots inventés utilisent des préfixes ou suffixes utilisés couramment pour créer des termes).

	CLEAR et ClassYN				GrandMed-Ro			
Échelle d'évaluation	**T5-base Exp 1**		**T5-base Exp 2**		**T5-base Exp 1**		**T5-base Exp 2**	
	N°	**%**	**N°**	**%**	**N°**	**%**	**N°**	**%**
Score 2	73	3,21%	50	2,20%	3	0,2%	7	0,47%
Score 1	244	10,75%	330	14,55%	117	7,85%	214	14,41%
Scores positifs	***317***	***13,97%***	***382***	***16,85%***	***120***	***8,05%***	***223***	***15,01%***
Score 0	1 848	81,48%	1 796	79,22%	1 320	88,59%	1 216	81,88%
Score -1	94	4,14%	89	3,92%	50	3,35%	45	3,03%
Scores négatifs	***1 942***	***85,62%***	***1 885***	***83,14%***	***1 370***	***91,94%***	***1 261***	***84,91%***
Total	**2 268**	**100%**	**2 267**	**100%**	**1 490**	**100%**	**1 485**	**100%**

TABLE 4 : Statistiques sur les scores de l'échelle d'évaluation de prédictions : expériences 1 et 2

Nous avons analysé les prédictions annotées avec le *score 1* afin d'identifier celles qui sont des nouvelles reformulations par rapport au jeu de données d'entraînement (les 8 146 paires *terme-reformulation* pour le français et les 2 727 paires pour le roumain). Nous analysons les prédictions uniques, sans doublons, et nous observons que lors de l'**Exp 2**, **T5-base** a généré **81,55%** de nouvelles reformulations en français et **86%** en roumain (Table 5). Les nouvelles reformulations obtenues sont en général des reformulations assez simples, des variantes du nom de la maladie sous forme d'adjectif (*schizophrénie : une maladie schizophrénique*).

	CLEAR et ClassYN				GrandMed-Ro			
Modèle	**T5-base Exp 1**		**T5-base Exp 2**		**T5-base Exp 1**		**T5-base Exp 2**	
	N° ref	**%**	**N° ref**	**%**	**N° ref**	**%**	**N° ref**	**%**
nouvelle prédiction	135	**68,52 %**	84	**81,55 %**	94	**84,68 %**	172	**86 %**
prédiction entraînement	62	31,47 %	19	18,44 %	17	15,31 %	28	14 %
Total sans doublons	**197**	**100 %**	**103**	**100 %**	**111**	**100 %**	**200**	**100 %**

TABLE 5 : Statistiques sur les prédictions automatiques de score 1

Les résultats des scores négatifs sont dus également aux mots inventés ou mal orthographiés, générés par le Transformer. En français nous avons identifié seulement 8 mots inventés (0,35%) de type « maladie *nichéolaire* », « une maladie caractérisée par une *ote* de cœur ». Cependant, pour le roumain le nombre de mots inventés est beaucoup plus grand (**Exp 1** : 71 mots (4,76%) ; **Exp 2** : 129 mots (8,68%)), comme nous l'observons dans les exemples suivants : « reprezinta o afectiune in care se dezvolta un *tumefl rea* in *cortija* » (*représente une affection dont se developpe [mot inventé] mauvaise [mot inventé]*), « numiti *articulatari* » (*nommés [mot inventé]*), « este o boala a *virusesului* » (*est une maladie de [mot inventé]*).

Nous remarquons que ces mots inventés ont un lemme correct, comme dans les deux derniers exemples, « articulație » (*articulation*) et « virus » (*virus*). La difficulté pour le roumain vient de la déclinaison avec article enclitique (attaché à la fin du mot) et par la présence des cas en roumain. Dans l'exemple « este o boala a *virusesului* », la forme correcte du dernier mot serait la forme articulée, cas génitif du mot *virus*, c'est-à-dire « a virusului » (*du virus*). La forme incorrecte « a virusesului » montre que le Transformer n'a pas trouvé la bonne particule à ajouter à la fin du mot pour illustrer le cas génitif (qui exprime la possession en roumain) ou qu'il a ajouté la particule au mot anglais *viruses* (forme au pluriel de *virus*). Nous observons ce problème également avec le mot « articulatari », dont la forme correcte est « articulații » (*articulations*). L'absence de signes diacritiques du roumain (ă, î, â, ț, ș), problème récurrent dans les textes de vulgarisation sur la toile, complexifie encore la tâche du Transformer.

D'autres erreurs de génération concernent l'insertion des mots ou séquences de mots qui n'ont pas de lien avec le contexte dans la reformulation. Le terme « cholangite sclérosante primitive » (« maladie intestinale inflammatoire » (reformulation de référence)) est reformulé comme « une maladie chronique qui peut être caractérisée par des troubles cholestatiques, des antécédents et des taches causés par une inflammation des voies et des structures de la peau ». « Les structures de la peau » n'ont pas de lien direct avec les maladies intestinales. Ce type d'erreur est difficilement identifiable mais apparaît fréquemment dans les paraphrases annotées avec le *score 0*.

4.4 Analyse de la lisibilité des prédictions

Nous avons analysé le niveau de lisibilité (Laframboise, 1978 ; François, 2011) des reformulations générées automatiquement. Nous avons constitué un guide d'annotation de la lisibilité pour les non-spécialistes sur trois niveaux :

- *Niveau 1, facile à comprendre* : la reformulation médicale est plus facile à comprendre que le terme médical (il y a que des mots plus simples dans la reformulation) ;

- *Niveau 2, même complexité* : même niveau de complexité ou de technicité entre le terme médical et sa reformulation, c'est-à-dire que le sens de deux parties est difficile à comprendre par l'annotateur ;
- *Niveau 3, difficile à comprendre* : la reformulation médicale est plus complexe ou plus technique que le terme et par conséquent plus difficile à comprendre.

L'évaluation sur les deux langues est réalisée par deux annotateurs non-spécialistes de la médecine, francophones ayant comme langue maternelle le roumain. Nous limitons cette analyse seulement aux nouvelles prédictions de reformulations (*score 1*). Nous n'avons pas évalué les reformulations incorrectes (*score 0*) à cause de la présence de termes inventés et de sens perturbés (mots inadaptés au contexte).

Niveau de lisibilité	**CLEAR et ClassYN**		**GrandMed-Ro**	
	Annot 1	**Annot 2**	**Annot 1**	**Annot 2**
	N° (%)	**N° (%)**	**N° (%)**	**N° (%)**
Niveau 1	221 (66,96%)	247 (74,84%)	179 (83,64%)	193 (90,18%)
Niveau 2	72 (21,81%)	42 (12,72%)	27 (12,61%)	2 (0,93%)
Niveau 3	37 (11,21%)	41 (12,42%)	8 (3,73%)	19 (8,87%)
Total	**330 (100%)**		**214 (100%)**	

TABLE 6 : Évaluation du niveau de lisibilité des prédictions

Les résultats de l'évaluation montrent que, en moyenne, **70,90%** des prédictions en français et **86,91%** en roumain sont plus faciles à comprendre que le terme médical. Ces prédictions peuvent servir à la simplification automatique de termes médicaux.

5 Conclusion

Nos expériences préliminaires prouvent que l'architecture **APT** peut être utilisée également pour générer des **reformulations sous-phrastiques médicales** avec une précision de **44,79%** pour le français et **45,66%** pour le roumain (si l'on considère les termes qui ont au moins une bonne prédiction). Si Nighojkar et Licato (2021) ont généré un grand nombre de paraphrases de la langue générale en anglais, nos expériences sont menées sur des données du domaine médical en français et en roumain, ce qui rend la tâche plus difficile. Les résultats sont comparables entre les deux langues.

Nous avons créé des **ressources** annotées et vérifiées manuellement partageables[3] et un **guide d'annotation et d'évaluation des reformulations** issues des textes naturels et des générations automatiques. Le guide peut s'appliquer pour l'évaluation d'autres jeux de données (termes et de leur reformulation) pour d'autres domaines. Nous nous analysé les nouvelles reformulations générées automatiquement et nous avons interprété les résultats. Notre adaptation d'un seul paramètre pour éviter les répétitions montre que l'architecture **APT** permet **d'augmenter** la précision (concernant le nombre de termes ayant au moins une reformulation correcte générée automatiquement) à **45%**. Nous exploiterons les erreurs observées afin de les éviter dans des expériences futures pour améliorer nos résultats de prédictions sur les deux langues d'étude, en particulier pour éviter les mots inventés. L'outil DERIF (Namer, 2002) pourrait être utilisé pour générer des paraphrases à partir des mêmes familles lexicales ou évaluer certaines reformulations générées automatiquement. L'évaluation manuelle des prédictions générées prouve qu'elles peuvent constituer des reformulations médicales faciles à comprendre par le grand public.

[3] Les ressources, le code et le guide seront disponibles à partir de 15 juin 2023 sur la plateforme github : https://github.com/ibuhnila/refomed

Références

AGIRRE E., BANEA C., CER D., DIAB M., GONZALEZ AGIRRE A., MIHALCEA R., RIGAU G. & WIEBE J. (2016). Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation*; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511. ACL (Association for Computational Linguistics).

ANTOINE E. & GRABAR N. (2016). Exploitation de reformulations pour l'acquisition d'un vocabulaire expert/non expert. In *TALN 2016 : Traitement Automatique des Langues Naturelles*. Paris, France. HAL : https://hal.archives-ouvertes.fr/hal-01426816.

BODENREIDER O. (2004). The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic Acids Research 32* (90001): 267D - 270. DOI : https://doi.org/10.1093/nar/gkh061.

BOWMAN S., GAUTHIER J., RASTOGI A., GUPTA R., MANNING C. D. & POTTS C. (2016). A Fast Unified Model for Parsing and Sentence Understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), p. 1466-1477.

BROCKETT C. & DOLAN W. B. (2005). Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, p. 1-8. http://aclweb.org/anthology/I/I05/I05-5001.

BUHNILA I. (2018). *Simplification lexicale entre les textes scientifiques et les textes de vulgarisation du domaine de la médecine.* Mémoire de Master. Université de Strasbourg, France.

BUHNILA I. (2021). Building a Corpus of Medical Paraphrases in Romanian. In *Proceedings of the The 16th Edition of the International Conference on Linguistic Resources and Tools for Natural Language Processing – ConsILR-2021*, Iasi, p. 139-152.

BUHNILA I. (2022a). Le Rôle Des Marqueurs et Indicateurs Dans l'analyse Lexicale et Sémantico-Pragmatique de Reformulations Médicales. *8e Congrès Mondial de Linguistique Française (CMLF)*, 4-8 juillet 2022, Orléans, SHS Web of Conferences 138: 10005. DOI : https://doi.org/10.1051/shsconf/202213810005.

BUHNILA I. (2022b). Identifying Medical Paraphrases in Scientific versus Popularization Texts in French for Laypeople Understanding. In *Proceedings of the Third Workshop on Scholarly Document Processing*. Gyeongju, Republic of Korea, p. 69-79. Association for Computational Linguistics.

CARDON R. (2018). Approche lexicale de la simplification automatique de textes médicaux. In *Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018*, p. 159-73. Rennes, France.

CARDON R. (2021). *Simplification automatique de textes techniques et spécialisés*. Informatique et langage [cs.CL]. Thèse de doctorat. Université de Lille. Français. ⟨NNT : 2021LILUH007⟩. ⟨tel-03343769v2⟩.

CARDON R. & GRABAR N. (2019). Automatic detection of parallel sentences in comparable biomedical corpora. In *TALN 2019*. Toulouse, France. HAL : https://hal.archives-ouvertes.fr/hal-02430446.

COHEN, J. (1960). A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.,* 20, p. 27-46.

CÔTÉ R. A. (1998). Systematized Nomenclature of Human and Veterinary Medicine: SNOMED International. Version 3.5. *Northfield, IL: College of American Pathologists*.

CONDAMINES A. (2018). Nouvelles perspectives pour la terminologie textuelle. J. Altmanova; M. Centrella; K.E. Russo. *Terminology and Discourse, Peter Lang*, p. 1-13.

COSTA R. (2005). Texte, terme et contexte. In *Actes des septièmes Journées scientifiques du réseau de chercheurs Lexicologie Terminologie Traduction*, p. 79-88. Bruxelles, Belgique.

DELÉGER L. & ZWEIGENBAUM P. (2009). Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora*, p. 2–10. BUCC '09. Suntec, Singapore: Association for Computational Linguistics.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*. http://arxiv.org/abs/1810.04805.

DONNELLY K. (2006). SNOMED-CT: The Advanced Terminology and Coding System for EHealth. *Studies in Health Technology and Informatics*, 121, p. 279-90.

ELHADAD N. & SUTARIA K. (2007). Mining a Lexicon of Technical Terms and Lay Equivalents. In *Biological, translational, and clinical language processing*, p. 49–56. Prague, Czech Republic: Association for Computational Linguistics. https://www.aclweb.org/anthology/W07-1007.

ESHKOL-TARAVELLA I. & GRABAR N. (2017). Taxinomie dans les reformulations du point de vue de la linguistique de corpus. *Syntaxe et Sémantique, vol. 18, no. 1*, p. 149-184.

FRANÇOIS T. (2011). *Les apports du traitement automatique du langage à la lisibilité du français langue étrangère*. Thèse de Doctorat. Université Catholique de Louvain. Louvain, France.

GRABAR N. & CARDON R. (2018). CLEAR - Simple Corpus for Medical French. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, p. 3-9. Tilburg, the Netherlands: Association for Computational Linguistics. DOI : https://doi.org/10.18653/v1/W18-7002.

GRABAR N. & HAMON T. (2015). Extraction automatique de paraphrases grand public pour les termes médicaux. In *22ème Traitement Automatique des Langues Naturelles*, 14. Caen, France.

GRABAR N. & HAMON T. (2016). Exploitation de la morphologie pour l'extraction automatique de paraphrases grand public des termes médicaux. *Traitement Automatique des Langues*, Varia, 57 (1), p. 85-109.

GUPTA A., AGARWAL A., SINGH P. & RAI P. (2018). A deep generative framework for paraphrase generation. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

KILGARRIFF A., BAISA V., BUŠTA J., JAKUBÍČEK M., KOVÁŘ V., MICHELFEIT J., RYCHLÝ P. & SUCHOMEL V. (2014). The Sketch Engine: ten years on. *Lexicography 1*, p. 7-36.

KOPTIENT A., CARDON R. & GRABAR N. (2019). Simplification-induced transformations: typology and some characteristics. In *BioNLP 2019*. Florence, Italy. DOI : https://doi.org/10.18653/v1/W19-5033.

LAFRAMBOISE Y. (1978). La lisibilité : Qu'est-ce que la lisibilité ? Quels éléments rendent un texte lisible et un autre pas ? *Québec français 32*, p. 27-29.

MALAISE V., ZWEIGENBAUM P. & BACHIMONT, B. (2004). Repérage et exploitation d'énoncés définitoires en corpus pour l'aide à la construction d'ontologie. In *Actes de la 11ème conférence sur le Traitement Automatique des Langues Naturelles*. Articles longs, p. 149–158, Fès, Maroc. ATALA.

MILLER G. A. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

MITROFAN M., BARBU MITITELU V. & MITROFAN G. (2019). MoNERo: A Biomedical Gold Standard Corpus for the Romanian Language. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, p. 71-79. Florence, Italy: Association for Computational Linguistics. DOI : https://doi.org/10.18653/v1/W19-5008.

NAMER, F. (2002). Acquisition automatique de sens à partir d'opérations morphologiques en français : études de cas. In *Actes de la 9ème conférence sur le Traitement Automatique des Langues Naturelles*. Articles longs, p. 237–246, Nancy, France. ATALA.

NIGHOJKAR, A. & LICATO, J. (2021). Improving Paraphrase Detection with the Adversarial Paraphrasing Task. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), p. 7106–7116, Online. Association for Computational Linguistics.

PECOUT A., TRAN T. M. & GRABAR N. (2019). Améliorer la diffusion de l'information sur la maladie d'Alzheimer : étude pilote sur la simplification de textes médicaux. *Ela. Etudes de linguistique appliquée*, no 195 (3), p. 325-41.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research (JMLR)*, 21(140).

RAMADIER L. (2016). Indexation et apprentissage de termes et de relations à partir de comptes rendus de radiologie. Thèse en Informatique. Université Montpellier, Français. ⟨NNT : 2016MONTT298⟩. ⟨tel-01479769v2⟩.

SAGGION H. (2017). Automatic Text Simplification. *Synthesis Lectures on Human Language Technologies* 10 (1), p. 1-137. DOI : https://doi.org/10.2200/S00700ED1V01Y201602HLT032.

SAPOIU C. (2013). *Hiponimia în terminologia medicală. Modalităţi de abordare în semantică şi lexicografie.* Piteşti, Editura Trend, 199 pages.

SELLAM T., DAS D. & PARIKH A. (2020). BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7881–7892, Online. Association for Computational Linguistics.

SPACKMAN, K. A., CAMPBELL K. E. & CÔTÉ, R. A. (1997). SNOMED RT: a reference terminology for health care. In *Proceedings of the AMIA Annual Fall Symposium*, p. 640-44.

SPECIA L., KUMAR J. S. & MIHALCEA R. (2012). SemEval-2012 task 1: English Lexical Simplification. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task*, and Volume 2: *Proceedings of the Sixth International Workshop on Semantic Evaluation*, p. 347–355. SemEval '12. Montréal, Canada: Association for Computational Linguistics.

SHARDLOW M. (2014). A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications* 4 (1). DOI : https://doi.org/10.14569/SpecialIssue.2014.040109.

TODIRASCU A., PADO S., KRISCH J., KISSELEW M. & HEID U. (2012). French and German Corpora for Audience-Based Text Type Classification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation* (LREC'12), p. 1591–1597. Istanbul, Turkey: European Language Resources Association (ELRA).

TCHECHMEDJIEV A., ABDAOUI A., EMONET V., ZEVIO S. & JONQUET C. (2018). SIFR annotator: ontology-based semantic annotation of French biomedical text and clinical notes. *BMC bioinformatics*, 19(1), 405.

VARGAS E. (2008). Un comportement de type céramique, c'est-à-dire cassant : les reformulations intratextuelles dans les émissions de vulgarisation télévisées allemandes. In *Pragmatique de la reformulation. Types de discours - Interactions didactiques*. Sous la direction de M. SCHUWER. M.-C. LE BOT, E. RICHARD, p. 21-38. Rennes: Presses Universitaires de Rennes.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

XU W., CALLISON-BURCH C. & DOLAN W. B. (2015). Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, p. 1-11.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Étude comparative des plongements lexicaux pour l'extraction d'entités nommées en français

Danrun Cao[1, 2] Nicolas Béchet [1] Pierre-François Marteau [1]
(1) Univ. Bretagne Sud, CNRS, IRISA, Rue Yves Mainguy, 56000 Vannes, France
(2) OctopusMind, 2 Pl. Saint-Pierre, 44000 Nantes, France
danrun.cao@univ-ubs.fr, nicolas.bechet@univ-ubs.fr,
pierre-francois.marteau@univ-ubs.fr

RÉSUMÉ

Dans ce papier nous présentons une étude comparative des méthodes de plongements lexicaux pour le français sur la tâche de Reconnaissance d'entités nommées (REN). L'objectif est de comparer la performance de chaque méthode sur la même tâche et sous les mêmes conditions de travail. Nous utilisons comme corpus d'étude la proportion française du corpus WikiNER. Il s'agit d'un corpus de 3,5 millions tokens avec 4 types d'entités. 10 types de plongements lexicaux sont étudiés, y compris les plongements non-contextuels, des contextuels et éventuellement ceux à base de transformer. Pour chaque plongement, nous entraînons un BiLSTM-CRF comme classifieur. Pour les modèles à base de transformer, nous comparons également leur performance sous un autre cas d'usage : *fine-tuning*.

ABSTRACT

Comparative study of word embeddings for French Named Entity Recognition

In this paper we present a comparative study of word embedding methods for French named entity recognition (NER). Our objectif is to compare each method's performance when facing the same task and under the same working conditions. We use the French proportion of WikiNER as corpus of study. It is a 3.5-million token corpus with 4 entity types. 10 embedding methods are studied, including non-contextual ones, contextual ones and transformer-based ones. For each embedding, we train a BiLSTM-CRF as token classifier. for transformer-based models, we also compare their performance under another usage which is fine-tuning.

MOTS-CLÉS : Plongements lexicaux, Reconnaissance d'entités nommées, état de l'art.

KEYWORDS: Word embeddings, Names entity recognition, benchmark.

1 Introduction

En traitement automatique des langues (TAL), une entité nommée est définie comme étant un syntagme nominal se référant à un objet réel du monde. Ce syntagme est appelé la "mention" de l'entité en question. Une entité peut avoir plusieurs mentions. Par exemple "*France*" et "*République Française*" sont deux mentions renvoyant à l'entité "*France*". Ces objets peuvent être abstraits ou concrets, en fonction des besoins de chaque cas d'application. Il est important que la mention permette aux lecteurs d'identifier l'entité cible de manière précise et unique. Ceci sous entend qu'une mention ne peut être associée qu'à une seule entité. Quelques exemples courants d'entités nommées correspondent à des noms de personnes ou d'entreprises, des endroits, des dates, des évènements, etc.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 94–104.

La tâche de reconnaissance d'entités nommées (REN) consiste à identifier ces entités dans un texte non structuré et à leur attribuer une liste de catégories prédéfinies. La tâche CoNLL 2003 (Sang & De Meulder, 2003) en est un exemple bien connu. Elle propose deux corpus annotés en anglais et en allemand. Quatre types d'entités sont traités : personne (PER, *person*), endroit (LOC, *location*), organisation (ORG, *organization*) et divers (MISC, *miscellaneous*). La REN est non seulement une tâche importante en soi, mais elle constitue aussi un composant clé pour des tâches plus complexes comme l'extraction de relations sémantiques, le résumé automatique, la fouille d'opinion, etc.

Historiquement, la tâche REN est abordée avec des méthodes à base de règles. Ces règles sont développées par des experts du domaine correspondant. Plusieurs pistes peuvent être suivies, par exemple en s'appuyant sur les caractéristiques morphologiques des entités (majuscule, minuscule, expressions régulières...), les informations lexicales et syntaxiques des mots, et éventuellement des ressources complémentaires (dictionnaires terminologiques, bases de connaissance...). Un des premiers travaux publiés de ce genre est (Rau, 1991), où l'auteure résout un problème de détection de noms d'entreprise par un système de règles symboliques. La méthode repose sur la mise en oeuvre de règles empiriques, et elle est peu coûteuse en temps de calcul. Cependant, elle est dépendante d'une intervention humaine importante et de ressources lexicales propres au domaine traité, ce qui constitue sa principale faiblesse. Ces ressources peuvent être coûteuses à produire. Il n'est pas toujours aisé d'y accéder, pour des raisons de propriétés intellectuelles et de moyen d'acquisition. Par ailleurs, les règles, ainsi que les ressources lexicales, nécessitent des mises à jour régulières, ce qui constitue un travail fastidieux.

Les approches plus récentes pour la REN privilégient les méthodes par apprentissage automatique. Elles prennent en entrée du texte brut, le vectorisent et utilisent un modèle de classification pour produire des annotations d'entité au niveau du token. On entend par "token" une séquence continue de caractères non vides, qui constitue l'unité minimale prise en compte lors du traitement automatique des données textuelles. Au début, ce vecteur de token se définit manuellement. Nous choisissons des caractéristiques et/ou des heuristiques permettant de capturer les spécificités d'un token, et leur valeur constitue alors le vecteur du token. Nous entraînons ensuite un annotateur automatique, tels que SVM et CRF, qui classifie les tokens en fonction de leur vecteur (Shen et al., 2004; Kazama & Torisawa, 2007).

Dans cette étude, nous nous intéressons à une méthode qui consiste à projeter les tokens dans un espace vectoriel continu, et ce de manière automatique. Cette méthode, que nous appellerons plongement lexical dans le reste de l'article, permettent d'encoder plus ou moins efficacement l'espace sémantique d'un token et éventuellement d'un texte entier. Cet article présente une étude comparative, pour la langue française, des méthodes de plongement issues de l'état de l'art sur une tâche de REN, en exploitant une même méthode de classification au niveau token. Cette étude permet d'analyser quantitativement l'impact des différents plongements lexicaux sur les résultats obtenus.

2 Travaux connexes

2.1 Plongement lexical

On entend par "plongement lexical" la représentation vectorielle des mots dans un espace métrique. Cet espace peut être interprété comme un espace sémantique de la langue, pour lequel chaque dimension correspond à un concept sémantique concret ou abstrait. La valeur des différentes dimensions décrit

alors la "proportion" de présence de chaque concept dans un mot, ce qui constitue une représentation quantifiée du "sens" du mot.

Nous distinguons deux types de plongements : non-contextuel et contextuel. Pour les plongements non-contextuels, chaque mot est associé à un vecteur unique, quel que soit le contexte dans lequel il est utilisé. Le premier algorithme de ce genre est *SENNA* (Collobert et al., 2011). *SENNA* initialise des vecteurs aléatoires pour chaque mot du vocabulaire et les entraîne de manière conjointe avec le classifieur en charge d'étiqueter les tokens. Il est possible d'ajouter des descripteurs empiriques comme ceux utilisés dans les systèmes à base de règles. Pour ce faire, *SENNA* génère des vecteurs aléatoires pour chacune des valeurs de ces descripteurs, et les concatène aux vecteurs de mots. Ces vecteurs de descripteurs sont également entraînés conjointement avec l'étiqueteur. Un peu plus tard, (Mikolov et al., 2013) ont proposé *Word2Vec*. Son concept clé est qu'un mot peut être défini par son contexte, et que les mots qui partagent un même contexte sont potentiellement des synonymes, lesquels pourront alors être représentés par des vecteurs proches. *fastText* (Bojanowski et al., 2017) est une extension de *Word2Vec* qui exploite les informations des sous-mots, c.-à-d. les n-grammes de caractères qui composent les mots. Cela permet notamment de prendre en compte les affixes et les racines des mots, et de mieux gérer les mots hors vocabulaire. Dans une autre approche, *GloVe* (Pennington et al., 2014) calcule les vecteurs en s'appuyant sur les co-occurrences des mots. Un dernier exemple de plongement non-contextuel est proposé dans (Chiu & Nichols, 2016). Ce modèle génère des vecteurs de mot par agrégation de vecteurs définis au niveau des caractères. L'ordre et la composition des caractères permettent également de représenter l'espace sémantique d'un mot. Chaque caractère initialement reçoit un vecteur aléatoire.

Les plongements non-contextuels ont montré de belles performances dans plusieurs tâches de TAL telles que l'analyse de sentiment, la fouille d'opinion, etc. Cependant ils ne traitent pas correctement les cas d'ambiguïté comme les polysèmes et les expressions figées. Pour corriger cette lacune, les plongements contextuels ont été introduits. Ils ont la particularité de travailler au niveau de la phrase plutôt qu'au niveau des mots isolés. *ELMo* (Peters et al., 2018) est un réseau de neurones constitué de deux couches de *Long Short-Term Memory (LSTM)* (Hochreiter & Schmidhuber, 1997) bidirectionnel (*BiLSTM*) qui traite une phrase vectorisée en entrée. *LSTM* est un réseau de neurones récurrents spécialisé dans le traitement des données séquentielles. En plus de sa capacité à "mémoriser" les entrées précédentes, il décide également à quel point cette mémoire doit être prise en compte dans l'étape courante. Un *biLSTM* est donc un *LSTM* qui considère la séquence de données dans les deux sens : vers l'avant (*forward*) et vers l'arrière (*backward*). Ce processus produit cinq représentations d'un mot : la suite de vecteurs de mots, 2 représentations *forward*, et 2 *backward*. Le vecteur final est produit par une combinaison pondérée de ces cinq représentations, les poids étant dépendants de la tâche à réaliser. Un autre exemple de plongement contextuel est *flair* (Akbik et al., 2018). *flair* considère une phrase comme une suite de caractères, et ce dans les deux sens. Un *LSTM* est entraîné à prédire le prochain caractère étant donné les précédents (ou les suivants). La représentation finale d'un mot est obtenue par concaténation de deux éléments : l'encodage du premier caractère de la phrase jusqu'au dernier caractère du mot, et celui du dernier caractère de la phrase jusqu'au début du mot.

Les architectures récentes à base de *Transformer* (Vaswani et al., 2017) se sont rapidement imposées dans l'état de l'art des méthodes de vectorisation. Une caractéristique importante du modèle Transformer est le mécanisme d'attention. Ce dernier permet au modèle de mesurer l'impact de chacun des autres tokens du texte lors du traitement du token courant, et ce en optimisant les poids d'attention. Le modèle tient compte également de l'ordre d'occurrence des tokens, en exploitant un vecteur qui encode l'emplacement de chaque token. Le premier modèle de plongement à base de *Transformer*

publié est *BERT* (Devlin et al., 2019). Il reprend la structure de Transformer d'origine, à savoir 12 couches identiques. Le modèle est entraîné sur deux tâches cibles non-supervisées : prédiction du token caché (*Masked Language Modeling*, *MLM*) et prédiction de la prochaine phrase (*Next Sentence Predictions*, *NSP*). Ces deux objectifs permettent au modèle de capturer à la fois les informations lexicales et phrastiques. Depuis, de nombreux variants de *BERT* ont été proposés, notamment :

— *RoBERTa* (Liu et al., 2019) et *DistilBERT* (Sanh et al., 2020) qui apportent des modification au niveau de la structure du réseau.

— *CamemBERT* (Martin et al., 2020), *FlauBERT* (Le et al., 2020) et *PhoBERT* (Nguyen & Tuan Nguyen, 2020) qui sont spécialisés sur d'autres langues que l'anglais.

À part *BERT* et les modèles dérivés, un autre modèle *Transformer*, *XLM* (Lample & Conneau, 2019), a été proposé. *XML* reprend également la structure de Transformer d'origine, mais en exploitant trois tâches d'entraînement : *MLM* comme dans *BERT*, prédiction du prochain token (*Causal Language Modeling*, *CLM*), et traduction (*Translation Language Modeling*, *TLM*). Un variant *XLM-R* (Conneau et al., 2020) basé sur l'architecture de *RoBERTa* a également été développé.

2.2 REN pour le français

En analysant les travaux effectués sur la REN en français, nous avons constaté un manque de corpus volumineux en français librement accessibles pour le développement des modèles de REN.

Le premier corpus REN en langue française librement accessible est proposé par (Sagot et al., 2012). Il s'agit de l'annotation d'entités nommées sur le corpus French Treebank (FTB) (Abeillé et al., 2003). Le corpus comprend 5 890 phrases issues du journal *Le Monde*, avec un total de 11 636 entités. Sur ce corpus, (Dupont, 2017) a entraîné et proposé l'outil SEM. La méthode s'appuie sur des descripteurs empiriques tels que les affixes, les noms communs précédents et suivants l'entité, ainsi que sur des lexiques. Un modèle à base de champ aléatoire conditionnel (*conditional random field* ou *CRF*, (Lafferty et al., 2001)) est ensuite entraîné puis utilisé comme classifieur. Ce système obtient un F1 score de 83,7% sur le corpus FTB-NER. (Suárez et al., 2020) ont proposé une comparaison entre SEM, *fastText*, *CamemBERT* et *FrELMo*. Le meilleur résultat est obtenu par la combinaison de *fastText* et *CamemBERT*, et le classifieur *LSTM-CRF*, le F1 score atteignant 90,25%. Les auteurs de *CamemBERT* ont également évalué leur modèle sur FTB-NER. Ils emploient *CamemBERT* en plongement et entraînent un LSTM-CRF. Le F1 score est de 89,55%.

Europeana Newspapers (Neudecker, 2016) est un autre corpus REN multilingue en accès libre, qui inclut le français, constitué d'articles de journaux numérisés avec des outils d'OCR. Cependant, l'OCR utilisé introduit de nombreuses erreurs et le corpus nécessite un travail de correction additionnel.

Enfin, FENEC (Millour et al., 2022) est un corpus contenant six genres de textes (prose, poésie, informations, encyclopédie, parole et multi-sources), annoté en entité nommée sous le schéma Quaero (Rosset et al., 2011). Ce corpus contient 11 149 tokens et 875 entités.

3 Corpus

Nous choisissons la proportion française du corpus WikiNER (Nothman et al., 2013) comme corpus d'étude, nous le désignerons WikiNER-fr dans la suite. WikiNER-fr comprend 134 092 phrases et 3.5 millions tokens, issus de plus de 61 000 articles Wikipédia. Il concerne les mêmes type d'entités que

ceux exploités dans CoNLL 2003, à savoir PER, LOC, ORG et MISC. Le tableau 1 montre le nombre de tokens pour chaque type d'entités dans le corpus.

Entité	PER	LOC	ORG	MISC
Nb. de tokens	129 978	155 565	45 443	81 594

TABLE 1 – Nb. de tokens pour chaque type d'entités dans le corpus

Les annotations de WikiNER ont été produites de manière semi-supervisée. D'abord les auteurs ont catégorisé manuellement environ 2 500 articles. Ensuite un classifieur est entraîné sur ce corpus initial puis exploité pour annoter le reste des pages. Parmi les trois modèles de classification comparées, le meilleur est la Régression logistique, avec un F1 score autour de 93% pour toutes les langues.

Puis la catégorie de chaque page Wikipédia est projetée sur les mentions associées aux hyperliens qui pointent vers la page en question. Par exemple, si la page "France" est annotée LOC, et si dans la phrase "Foucault retourne en France en 1960", le mot "France" reçoit un hyperlien vers la page "France", alors le mot "France" sera étiqueté LOC. Dans un article Wikipédia, seule la première occurrence d'une entité reçoit un hyperlien. Les auteurs ont alors mis en place quatre stratégies d'inférence afin de repérer les autres mentions des entités dans le corpus. 5 variants du corpus sont proposés, du corpus le plus bruité/exhaustif au corpus le moins bruité/exhaustif. Dans notre étude, nous avons choisi le variant wiki-2, un bon compromis produit en considérant l'avant-dernier niveau d'inférence (Nothman et al., 2013).

A noter que, mis à part le corpus d'entraînement initial, aucune vérification manuelle n'a été réalisée sur l'ensemble du corpus. Le corpus est alors dit en version *silver-standard*. Les auteurs ont montré que les modèles entraînés sur un corpus *silver-standard* pouvaient atteindre une performance comparable, ou parfois mieux dans certains cas, à ceux appris sur un corpus *gold-standard*, c.-à-d. sur un corpus révisé manuellement. Or cette comparaison a été faite uniquement sur le sous-corpus anglais. Dans le but de vérifier cette propriété pour le Français, nous avons choisi aléatoirement 20% des phrases du WikiNER-fr et révisé manuellement les annotations. Le processus de révision est détaillé dans (ref anonyme). Nous appellerons ce corpus révisé WikiNER-fr-gold dans la suite.

4 Expériences

4.1 Plongements lexicaux non-contextuels

Parmi les plongements lexicaux non-contextuels, nous avons évalué *Word2Vec*, *fastText* et *GloVe*. Pour *Word2Vec*, nous avons exploité le modèle pré-entraîné par Jean-Philippe Fauconnier. Ce modèle a été entraîné sur le corpus frWaC (Baroni et al., 2009), un ensemble de textes issus du WEB constitué de 1,6 milliards tokens. Pour *Glove*, nous avons entraîné un nouveau modèle sur le corpus frWac, afin que les résultats puissent être comparés à ceux obtenus par *Word2Vec*. Nous le mettrons à disposition de la communauté. Finalement pour fastText, nous utilisons le modèle proposé par l'équipe développeuse pré-entraîné sur le corpus Common Crawl. Nous étudions également la contribution des n-grammes de caractères de *fastText* pour la REN.

4.2 Plongments lexicaux contextuels

7 types de plongements lexicaux contextuels sont évalués, dont cinq exploitent l'architecture de type *transformer*. Les 2 modèles *non-transformer* sont *flair* et *ELMo*. Pour *flair*, 3 modèles sont évalués : un modèle français entraîné sur Wikipédia français, un autre modèle français entraîné sur frWaC, et un modèle multilingue entraîné sur le corpus JW300 (Agić & Vulić, 2019). Ce dernier est un corpus parallèle de plus de 300 langues, dont le français. Concernant *ELMo*, il existe un modèle français pré-entraîné (Che et al., 2018), disponible dans la librairie ELMoForManyLangs.

Parmi les 5 modèles transformers, nous distinguons là encore deux catégories : monolingue et multilingue. Les modèles français évalués sont *CamemBERT* et *DistilCamemBERT* (Delestre & Amar, 2022). Ce dernier est un travail inspiré par *DistilBERT* appliqué sur *CamemBERT*. Les trois autres modèles multilingues sont : *BERT* multilingue, *DistilBERT* multlingue, et *XLM-R*. Certains de ces modèles proposent des variants entraînés sur différents corpus ou avec des nombres différents de paramètres. Nous évaluons tous ces variants mais ne présentons que les résultats du meilleur variant dans la section 5. Il existe deux cas d'usage courants des modèles transformers. Le premier consiste à rajouter une couche de type perceptron comme classifieur, puis à poursuivre l'entraînement d'un modèle pré-entraîné sur la tâche cible. Ce processus est appelé *fine-tuning*. Cela permet au modèle d'optimiser ses paramètres spécifiquement pour la tâche cible. Une fois *fine-tuné*, le modèle pourra être exploité sans tenir compte de la couche de classification. Le deuxième correspond aux situations dans lesquelles un modèle pré-entraîné sert uniquement à produire des vecteurs de mot statiques. Ces vecteurs sont utilisés pour entraîner un classifieur sur les deux cas d'usage que nous comparons.

4.3 Procédure

Pour chaque plongement, nous entraînons un biLSTM-CRF et l'utilisons pour extraire et catégoriser les entités nommées en quatre classes. Cette procédure est implémentée à l'aide de la librairie Flair (Akbik et al., 2019). Le réseau biLSTM est initialisé avec 512 neurones (configuration par défaut de Flair). Le classifieur est entraîné sur un nombre d'époques inférieur à 150. Le corpus est découpé en trois sous ensembles : 60% entraînement, 20% validation et 20% test. Le jeu de test contient les mêmes phrases que WikiNER-fr-gold. Lors de l'entraînement, le modèle apprend sur les données d'entraînement et est évalué sur les données de validation. Une fois que le modèle a convergé, il est évalué d'abord sur les données de test puis sur WikiNER-fr-gold. Ceci nous permet de mesurer l'impact des erreurs du corpus sur la performance des modèles. Le taux d'apprentissage est initialisé à $0,1$. Celui-ci diminue en fonction de la qualité de l'entraînement obtenue à chaque époque.

A part le modèle fastText qui inclut les n-grammes, les plongements non-contextuels sont tous confrontés au problème des mots inconnus. Il est ainsi nécessaire de définir manuellement le vecteur à attribuer aux mots inconnus. Nous comparons ici deux stratégies pour proposer un vecteur "par défaut" pour les termes inconnus : (1) nous considérons la moyenne de tous les vecteurs du modèle ou (2) nous utilisons un vecteur nul, i.e. un vecteur dont toutes les dimensions prennent la valeur 0.

Certains des modèles à évaluer, comme ELMo et BERT, sont par nature des réseaux de neurones qui n'ont pas de couche de sortie permettant de produire des plongements directement. Ce sont les couches cachées qui encodent la langue. Il faut donc choisir une stratégie d'utilisation de ces couches cachées. Pour ELMo, nous concaténons l'état caché des 3 couches, c.-à-d. la couche d'entrée et les 2 couches de BiLSTM. Selon les auteurs (Peters et al., 2018), l'utilisation de la concaténation des couches permet une meilleure performance en général que d'une seule couche spécifique. Pour les

modèles à base de transformers, nous nous sommes référés aux résultats présentés dans l'article d'origine de BERT (Devlin et al., 2019). Les auteurs ont testé plusieurs stratégies sur le benchmark CoNLL 2003, et le meilleur résultat est obtenu avec la concaténation des états cachés des 4 dernières couches. Nous reprenons cette même configuration dans nos expériences.

L'évaluation finale est effectuée aux niveaux token et entité. Au niveau token, nous vérifions si la bonne catégorie d'entité a été attribué à un token. Au niveau entité, il faut que l'annotation et les frontières soient correctes. Nous utilisons la précision, le rappel et le F1 score comme métrique d'évaluation, et nous calculons la moyenne pondérée de ces métriques pour les quatre types d'entité.

5 Résultats

Le tableau 5 présente nos résultats. Le meilleur résultat pour chaque type de plongement est en gras, et le meilleur résultat sur l'ensemble des plongements est donné en bas du tableau en gras italique.

Nous remarquons en premier lieu une perte de performance générale quand nous évaluons les modèles sur WikiNER-fr-gold, avec une baisse de 1,11% en moyenne au niveau token et 1,85% au niveau entité. La seule exception est mBERT qui améliore légèrement sa performance. Les modèles étant entraînés sur le corpus *silver-standard*, cette perte est attendue car les modèles apprennent sur des données bruitées. Dans les annotations produites par le meilleur modèle de chaque catégorie, nous constatons les mêmes types d'erreurs et d'incohérences que ceux présents dans le corpus d'origine. Les modèles sont capables d'en corriger quelques unes, mais tout en produisant d'autres erreurs. Ceci confirme la nécessité d'une correction de corpus que nous avons décrit et initié dans cette étude.

Parmi les plongements non-contextuels, le meilleur résultat est obtenu par fastText avec les n-grammes. Il atteint un niveau de performance supérieur même à certains modèles à base de transformer. La deuxième place est occupée avec un petit écart par sa version sans n-grammes qui utilise un vecteur nul pour représenter les mots hors vocabulaire. L'ajout des n-grammes a exigé 5 fois plus de RAM (10,3Go avec n-grammes contre 2,05Go sans n-grammes), sans produire une amélioration considérable (+0.23% en moyenne). Concernant les deux stratégies de gestion de mots hors vocabulaire, le vecteur moyenne semble être une meilleure option pour Word2Vec (+1,03% en moyenne) et notamment GloVe (+8,3% en moyenne). En revanche, il semble préférable d'utiliser un vecteur nul pour les modèles fastText (+6,1% en moyenne).

Dans les deux cas d'usage des plongements à base de transformer, c'est CamemBERT en version large qui produit les meilleurs F1 scores, suivi par CamemBERT en version base et XLM-R en version large. Parmi les variants de CamemBERT-base, celui pré-entraîné sur Wikipédia obtient le meilleur résultat en mode statique, et celui sur CCNet (Wenzek et al., 2020) semble meilleur en mode *fine-tuning*. Toutefois, nous remarquons que l'écart entre CamemBERT-large et CamemBERT-base n'est pas très grand (0.5% en statique et 0.1% en *fine-tuning*). Sachant que la taille de CamemBERT-large est trois fois celle de CamemBERT-base (350m paramètres contre 110m), l'entraînement ainsi que l'exploitation du premier exigent bien plus de ressources de calcul que le second. Ceci pourrait être un facteur important à prendre en compte pour les cas d'usage industriels par exemple où on voudrait limiter le coût de ressources. Dans le même esprit, nous avons comparé mBERT avec DistilBERT-m. Malgré un nombre de paramètres plus faible (-40% d'après les auteurs), DistilBERT-m obtient quasiment la même performance que mBERT.

Plongement	WikiNER-fr test		WikiNER-fr-gold	
	F1 token	F1 entité	F1 token	F1 entité
Plongements non-contextuels				
Word2Vec+moyenne	89,6%	87,4%	88,2%	84,2%
GloVe+moyenne	89,3%	87,3%	87,9%	83,9%
fastText+moyenne	87,8%	84,8%	86,7%	82%
Word2Vec+nul	87,8%	86,1%	86,9%	84,5%
GloVe+nul	81,9%	76,9%	80,9%	75,5%
fastText+nul	92,7%	91,3%	91,7%	89,8%
fastText+n-grammes	**92,9%**	**91,6%**	**91,9%**	**90%**
Plongements contextuels non-transformer				
ELMo	90,6%	88,8%	89,7%	87,3%
flair français Wikipédia	91,0%	89,1%	90,1%	87,7%
flair français frwac	**91,9%**	**90,2%**	**90,6%**	**88,3%**
flair multilingue	89,7%	87,5%	88,7%	86,2%
Plongements transformer français statiques				
Camembert base ccnet	93,9%	92,6%	92,5%	90,5%
Camembert base wikipédia	94,3%	93,0%	93.2%	91,2%
Camembert large	**95%**	**93,7%**	**93,5%**	**91,4%**
DistilCamembert base	92,7%	91,1%	91,4%	89,2%
Plongements transformer multilingues statiques				
mBERT	93,0%	91,5%	91,7%	89,6%
DistilBERT-m	92,7%	91,2%	91,5%	89,4%
XLM-R base	91,1%	89,3%	89,9%	87,6%
XLM-R large	**93,9%**	**92,5%**	**92,5%**	**90,5%**
Plongements transformer français + fine-tuning				
Camembert base ccnet	94,0%	92,3%	92,8%	90,3%
Camembert large	**94,1%**	**92,5%**	**92,8%**	**90,5%**
DistilCamembert base	93,1%	91,0%	92%	89,3%
Plongements transformer multilingues + fine-tuning				
mBERT	92,0%	89,6%	92,4%	89,7%
DistilBERT-m	92,5%	90,1%	91,3%	88,2%
XLM-R base	92,5%	90,5%	91,3%	88,5%
XLM-R large	**93,6%**	**91,6%**	**92,6%**	**90%**
Meilleur résultat				
Camembert large en statique	***95%***	***93,7%***	**93,5%**	**91,4%**

TABLE 2 – Résultats d'évaluation des plongements

6 Conclusion

Nous avons évalué 10 types de plongements lexicaux sur le corpus WikiNER-fr. L'objectif est de montrer, face à une même tâche et dans les mêmes conditions initiales, quel modèle est le plus adapté à une tâche REN. Nous avons effectué une évaluation supplémentaire des modèles sur WikiNER-fr-gold, la version révisée d'une partie du corpus WikiNER-fr. Dans les deux cas, les modèles à base de transformer ont obtenu les meilleurs résultats. Nous constatons une perte de performance générale lors de l'évaluation sur WikiNER-fr-gold. Les erreurs constatées dans le corpus WikiNER-fr sont reproduites dans les annotations réalisées par les modèles entraînés sur ce corpus. Il est donc nécessaire de réviser le corpus. Enfin, il convient de noter qu'un nombre de paramètres important ne garantit pas un gain de performance signifiant.

Références

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). Building a Treebank for French. In A. ABEILLÉ, Éd., Treebanks : Building and Using Parsed Corpora, volume 20, p. 165–187. Dordrecht : Springer Netherlands. DOI : 10.1007/978-94-010-0201-1_10.

AGIĆ & VULIĆ I. (2019). JW300 : A Wide-Coverage Parallel Corpus for Low-Resource Languages. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, p. 3204–3210, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1310.

AKBIK A., BERGMANN T., BLYTHE D., RASUL K., SCHWETER S. & VOLLGRAF R. (2019). FLAIR : An Easy-to-Use Framework for State-of-the-Art NLP. In NAACL 2019, p. 54–59.

AKBIK A., BLYTHE D. & VOLLGRAF R. (2018). Contextual String Embeddings for Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics, p.1̃2, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

BARONI M., BERNARDINI S., FERRARESI A. & ZANCHETTA E. (2009). The WaCky wide web : a collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation, **43**(3), 209–226. DOI : 10.1007/s10579-009-9081-4.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, **5**, 135–146. DOI : 10.1162/tacl_a_00051.

CHE W., LIU Y., WANG Y., ZHENG B. & LIU T. (2018). Towards Better UD Parsing : Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation. In Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies, p. 55–64, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/K18-2005.

CHIU J. P. C. & NICHOLS E. (2016). Named Entity Recognition with Bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, **4**, 357–370. DOI : 10.1162/tacl_a_00104.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural Language Processing (almost) from Scratch. J. Mach. Learn. Res., **12**, 2493–2537. DOI : 10.5555/1953048.2078186.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, p. 8440–8451 : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.747.

DELESTRE C. & AMAR A. (2022). DistilCamemBERT : a distillation of the French model CamemBERT. In CAp (Conférence sur l'Apprentissage automatique) : arXiv. DOI : 10.48550/ARXIV.2205.11111.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, p. 4171–4186, Minneapolis, Minnesota : ACL. DOI : 10.18653/v1/N19-1423.

DUPONT Y. (2017). Exploration de traits pour la reconnaissance d'entités nommées du Français par apprentissage automatique. In Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles. 19es REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL 2017), p. 42–55, Orléans, France : ATALA.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long Short-Term Memory. Neural Computation, **9**(8), 1735–1780. DOI : 10.1162/neco.1997.9.8.1735.

KAZAMA J. & TORISAWA K. (2007). Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In Proceedings of the 2007 EMNLP-CoNLL, p. 698–707, Prague, Czech Republic : Association for Computational Linguistics.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, p. 282–289, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc. DOI : 10.5555/645530.655813.

LAMPLE G. & CONNEAU A. (2019). Cross-lingual Language Model Pretraining. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, p.Ĩ1, Red Hook, NY, USA : Curran Associates Inc. DOI : 10.5555/3454287.3454921.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020). FlauBERT : Unsupervised Language Model Pre-training for French. In Proceedings of the Twelfth Language Resources and Evaluation Conference, p. 2479–2490, Marseille, France : European Language Resources Association.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, p. 1218–1227, Huhhot, China : Chinese Information Processing Society of China.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE , SEDDAH D. & SAGOT B. (2020). CamemBERT : a Tasty French Language Model. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, p. 7203–7219 : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. DOI : 10.48550/ARXIV.1301.3781.

MILLOUR A., DUPONT Y., JOUGLAR A. & FORT K. (2022). FENEC : un corpus à échantillons équilibrés pour l'évaluation des entités nommées en français. In Actes de la 29e Conférence sur

le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale, p. 82–94, Avignon, France : ATALA.

NEUDECKER C. (2016). An Open Corpus for Named Entity Recognition in Historic Newspapers. In Proceedings of the Tenth International Conference on Language Resources and Evaluation, p. 4348–4352, Portorož, Slovenia : European Language Resources Association.

NGUYEN D. Q. & TUAN NGUYEN A. (2020). PhoBERT : Pre-trained language models for Vietnamese. In Findings of the Association for Computational Linguistics : EMNLP 2020, p. 1037–1042 : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.92.

NOTHMAN J., RINGLAND N., RADFORD W., MURPHY T. & CURRAN J. R. (2013). Learning multilingual named entity recognition from Wikipedia. Artificial Intelligence, **194**, 151–175. DOI : 10.1016/j.artint.2012.03.006.

PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global Vectors for Word Representation. In Proceedings of EMNLP 2014, p. 1532–1543, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1162.

PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. In Proceedings of the 2018 NAACL : Human Language Technologies, Volume 1 (Long Papers), volume 1, p. 2227–2237, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1202.

RAU L. (1991). Extracting company names from text. In [1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application, volume i, p. 29–32, Miami Beach, FL, USA : IEEE Comput. Soc. Press. DOI : 10.1109/CAIA.1991.120841.

ROSSET S., GROUIN C. & ZWEIGENBAUM P. (2011). Entités nommées structurées : guide d'annotation Quaero.

SAGOT B., RICHARD M. & STERN R. (2012). Annotation référentielle du Corpus Arboré de Paris 7 en entités nommées. In Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2, p. 535–542, Grenoble, France : ATALA/AFCP.

SANG E. F. T. K. & DE MEULDER F. (2003). Introduction to the CoNLL-2003 Shared Task : Language-Independent Named Entity Recognition. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, p. 142–147.

SANH V., DEBUT L., CHAUMOND J. & WOLF T. (2020). DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter. DOI : 10.48550/ARXIV.1910.01108.

SHEN D., ZHANG J., SU J., ZHOU G. & TAN C.-L. (2004). Multi-criteria-based active learning for named entity recognition. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 589–es, Barcelona, Spain : Association for Computational Linguistics. DOI : 10.3115/1218955.1219030.

SUÁREZ P. J. O., DUPONT Y., MULLER B., ROMARY L. & SAGOT B. (2020). Establishing a New State-of-the-Art for French Named Entity Recognition. In Proceedings of the 12th LREC, p. 4631–4638, Marseille, France : European Language Resources Association.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, p. 6000–6010, Long Beach, California, USA : Curran Associates Inc. DOI : 10.5555/3295222.3295349.

WENZEK G., LACHAUX M.-A., CONNEAU A., CHAUDHARY V., GUZMÁN F., JOULIN A. & GRAVE E. (2020). CCNet : Extracting High Quality Monolingual Datasets from Web Crawl Data. In Proceedings of the 12th LREC, p. 4003–4012 : European Language Resources Association.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

"Honey, Tell Me What's Wrong", Explicabilité Globale des Modèles de TAL par la Génération Coopérative

Antoine Chaffin[1, 2*] Julien Delaunay[3*]
(1) IRISA, 263 Avenue du Général Leclerc, 35000 Rennes, France
(2) IMATAG, 13 Rue Dupont-des-Loges, 35000 Rennes, France
(3) Inria, 263 Avenue du Général Leclerc, 35000 Rennes, France
*Désigne une contribution équivalente
antoine.chaffin@irisa.fr, julien.delaunay@inria.fr

RÉSUMÉ

L'omniprésence de l'apprentissage automatique a mis en lumière l'importance des algorithmes d'explicabilité. Parmi ces algorithmes, les méthodes agnostiques au type de modèle génèrent des exemples artificiels en modifiant légèrement les données originales. Elles observent ensuite les changements de décision du modèle sur ces exemples artificiels. Cependant, de telles méthodes nécessitent d'avoir des exemples initiaux et fournissent des explications uniquement sur la décision pour ces derniers. Pour répondre à ces problématiques, nous proposons *Therapy*, la première méthode d'explicabilité modèle-agnostique pour les modèles de langue qui ne nécessite pas de données en entrée. Cette méthode génère des textes qui suivent la distribution apprise par le classifieur à expliquer grâce à la génération coopérative. Ne pas dépendre d'exemples initiaux permet, en plus d'être applicable lorsqu'aucune donnée n'est disponible (e.g, pour des raisons de confidentialité), de fournir des explications sur le fonctionnement global du modèle au lieu de plusieurs explications locales, offrant ainsi une vue d'ensemble du fonctionnement du modèle. Nos expériences montrent que, même sans données en entrée, Therapy fournit des informations instructives sur les caractéristiques des textes utilisées par le classifieur qui sont compétitives avec celles fournies par les méthodes utilisant des données.

ABSTRACT

"Honey, Tell Me What's Wrong", Global Explainability and Diagnosing of NLP Models through Cooperative Generation

The ubiquity of complex machine learning has raised the importance of model-agnostic explanation algorithms. These methods sample artificial instances by slightly perturbing target instances and observing the variations in the model decision. However, such methods require access to initial samples and only provide explanations of the decision for these. To tackle these problems, we propose Therapy, the first model-agnostic explanation method adapted to text which requires no input dataset. This method generates texts following the distribution learned by a classifier through cooperative generation. Not relying on initial samples, in addition to allowing use in cases where no data is available (e.g, for confidentiality reasons), provides global explanations of the model rather than multiple local ones, offering an overview of the model behavior. Our experiments show that although no input data is used to generate samples, Therapy provides insightful information about features used by the classifier that are competitive with the ones from methods relying on input samples.

MOTS-CLÉS : explicabilité, génération coopérative, traitement automatique des langues.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 105–122.

KEYWORDS: explainability, cooperative generation, natural language processing.

1 Introduction

L'émergence des modèles basés sur l'apprentissage automatique à permis leur adoption dans des domaines variés, allant de la simple recommandation à des secteurs critiques tels que la santé (Buch *et al.*, 2018; Esteva *et al.*, 2017; Karatza *et al.*, 2021) et le droit (Araszkiewicz *et al.*, 2022; Tagarelli & Simeri, 2022). Le besoin grandissant de précision induit une augmentation de la complexité de ces modèles, accentuant leur dénomination de boite noire. Ce manque de transparence limite (et empêche) leur déploiement dans différents domaines, en raison par exemple de l'augmentation significative de modèles souffrant de biais. Entre autres, certains *chatbots* ont été déployés bien que comportant des biais liés aux minorités religieuses (Abid *et al.*, 2021) ou de genre (Lucy & Bamman, 2021) et expliquer leur fonctionnement reste un problème ouvert.

Parmi les méthodes proposées pour répondre à ces problèmes, celles qui sont modèle-agnostiques sont préférées car applicables à tout type de modèle d'apprentissage automatique. Parmi ces dernières, les explications locales ont obtenu un fort succès car elles fournissent un bon compromis entre précision et utilité des explications. Les explications locales sont générées en perturbant une entrée afin de construire un voisinage autour de cette entrée et en étudiant comment le modèle réagit à ces petites différences. Cela permet de mettre en évidence les caractéristiques importantes pour le modèle et de fournir des éléments d'explication sur la décision du modèle pour cette entrée (e.g., les mots les plus importants de chaque classe). Selon une étude récente sur les tendances dans le domaine de l'explicabilité (Jacovi, 2023), les méthodes d'explication locale agnostique au modèle, telles que LIME (Ribeiro *et al.*, 2016) sont les plus utilisées.

Cependant, l'explication d'un modèle à partir d'un exemple précis présente trois défauts. Premièrement, il faut évidemment disposer d'entrées à expliquer ; ce qui peut être impossible pour des raisons de confidentialité ou de respect de la vie privée par exemple. Deuxièmement, choisir des entrées qui sont représentatives du modèle et/ou des données sur lesquelles le modèle sera utilisé est une tâche complexe. Troisièmement, cela donne une explication de la décision **pour cette entrée** et pour cette entrée uniquement. Ainsi, cela ne fournit que des informations très locales sur le comportement du modèle, qui ne représentent qu'une petite partie du domaine d'entrée du modèle. Pour cette raison, certaines méthodes d'explications ont proposé d'agréger plusieurs explications locales afin de fournir une explication plus globale. Néanmoins, ces explications restent fortement liées aux données d'entrée et ne fournissent des informations que sur le voisinage de ces échantillons. Ces méthodes nécessitent donc que les différentes données d'entrée couvrent la partie de l'espace la plus large possible.

Avec l'objectif de supprimer cette dépendance aux données d'entrée et de générer une explication globale du modèle, nous proposons ***Therapy***, une méthode qui utilise la génération coopérative (Holtzman *et al.*, 2018; Scialom *et al.*, 2020; Chen *et al.*, 2020; Bakhtin *et al.*, 2021; Chaffin *et al.*, 2022) pour générer des textes suivant la distribution d'un classifieur. La distribution des textes générés permet ensuite d'étudier les caractéristiques importantes du modèle, fournissant ainsi une explication globale agnostique au modèle. Dans ce papier, nous présentons d'abord les travaux connexes dans la Section 2, la Section 3 introduit ensuite des notions sur la génération de textes et plus particulièrement la génération coopérative. La Section 4 détaille quant à elle le fonctionnement de la méthode tandis que la Section 5 présente les expériences réalisées pour comparer les performances de Therapy avec les méthodes d'explication usuelles.

2 Travaux connexes

Générer une explication pour des données textuelles est une tâche ardue qui nécessite de prendre en compte à la fois la sémantique du texte mais également le domaine de la tâche (e.g. analyse de sentiment, détection de spam). De plus, il est fréquent de devoir évaluer un modèle déjà déployé pour des raisons d'équité ou de détection de biais par exemple mais que les données ne soient plus accessibles pour des raisons de sécurité ou de confidentialité. Afin de résoudre ce problème, les chercheurs se sont concentrés sur les méthodes d'explications post-hoc (Jacovi, 2023). Suivant la catégorisation de Bodria *et al.* (2021), nous différencions les explications sous forme d'exemples de celles par attribution de caractéristiques.

2.1 Explications sous forme d'exemples

Les méthodes d'explications sous forme d'exemples tirent leur racine des sciences sociales (Miller, 2019) et montrent des contrefactuels qui indiquent le changement minimum requis pour modifier une prédiction, ou des prototypes, des exemples représentatif d'une classe. Les méthodes contrefactuelles perturbent le document cible jusqu'à trouver le document le plus proche qui soit classé différemment par le modèle complexe. À l'inverse, les méthodes qui génèrent des prototypes sélectionnent les instances qui représentent le plus une classe cible. Parmi les méthodes *post-hoc*, certaines proposent des codes de contrôle permettant de surveiller la perturbation du texte en entrée, tandis que d'autres entraînent des mécanismes complexes pour générer des phrases réalistes en perturbant une instance dans un espace latent. Polyjuice (Wu *et al.*, 2021) et GYC (Madaan *et al.*, 2021) feront partie de la première catégorie et proposent des codes de contrôle allant du changement de sentiment ou de temps jusqu'à l'ajout ou le retrait de mots. xSPELLS (S. Punla *et al.*, 2022) et CounterfactualGAN (Robeer *et al.*, 2021), sont des méthodes qui entraînent respectivement un auto-encodeur variationnel et un réseau adversarial génératif pour convertir les textes en entrée dans un espace latent. Des modifications y sont ensuite réalisées afin de retourner des contrefactuels réalistes proches de l'exemple original.

2.2 Explications par attribution

Parmi les méthodes post-hoc, les explications par attribution associent un poids aux termes en entrée afin d'indiquer leur impact positif ou négatif sur la prédiction finale. Les méthodes telles que SHAP (Lundberg & Lee, 2017), LIME (Ribeiro *et al.*, 2016) et ses variantes (Gaudel *et al.*, 2022; Shankaranarayana & Runje, 2019; Zafar & Khan, 2019; Visani *et al.*, 2020; ElShawi *et al.*, 2019; Bramhall *et al.*, 2020) restent les plus utilisées pour générer des explications (Jacovi, 2023). Les explications sont dites locales puisque ces méthodes perturbent un document en entrée en modifiant légèrement les valeurs et en observant le comportement du modèle complexe dans cette localité. Pour les données texte, LIME masque aléatoirement les mots du document en entrée afin d'en créer diverses variations et entraîne un modèle linéaire sur ces exemples. Les coefficients du modèle linéaire, associés aux différents mots, sont ensuite retournés et utilisés comme explication. Bien que la majorité des études (Arrieta *et al.*, 2020; Bodria *et al.*, 2021) fasse une différence entre les explications locales et globales, LIME introduit LIME-SP (pour sélection sous modulaire), une méthode qui génère une explication globale à partir de n explications locales. Ces n explications sont choisies parmi un plus grand ensemble afin de couvrir le plus possible l'espace d'entrée tout en réduisant la redondance.

3 Génération de textes

3.1 Génération coopérative

Les modèles de langue génératifs (LM) tels que la famille des GPT (Radford *et al.*, 2018, 2019; Brown *et al.*, 2020) apprennent la distribution de probabilité de séquences de symboles $x_1, x_2, \cdots, x_T$ (souvent appelés *tokens*) appartenant à des séquences de taille variable T sur un vocabulaire $\mathcal{V}$. La probabilité d'une séquence x (aussi appelée vraisemblance) est définie comme la probabilité jointe de chacun de ses tokens. Cette probabilité peut être factorisée en utilisant la formule des probabilités composées : $p(x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{1:t-1})$. Le LM est entraîné à produire une distribution de probabilité sur le dictionnaire pour le prochain token sachant ceux en entrée, i.e, $p_\theta(x_t \mid x_{1:t-1})$ à un pas de temps donné t. Cela permet d'obtenir un modèle de langue auto-régressif qui génère des séquences itérativement. Le modèle utilise les distributions apprises pour émettre un token x_t et l'ajouter au contexte $x_{1:t-1}$, qui sera utilisé pour la prochaine itération. Le processus de génération –ou décodage– démarre généralement en utilisant une petite séquence initiale appelée l'amorce. Les grands modèles de langue apprennent une très bonne approximation de la distribution de leurs données d'apprentissage, ce qui permet de générer des textes plausibles en maximisant la vraisemblance du modèle $p(x)$. Cependant, cette approche offre très peu de contrôle sur le texte généré à l'exception de l'amorce.

Les approches de génération coopérative (Holtzman *et al.*, 2018; Scialom *et al.*, 2020; Chen *et al.*, 2020; Bakhtin *et al.*, 2021; Chaffin *et al.*, 2022) permettent de résoudre ce problème en utilisant des modèles discriminatifs pour guider le modèle de langue durant la génération. Elles utilisent l'information du modèle externe pour guider le modèle de langue afin de générer des textes qui possèdent une propriété reconnue par le modèle discriminatif. Dans le cas où le modèle est un classifieur qui apprend à prédire la probabilité $D(c \mid x)$ qu'une séquence x d'appartenir à la classe c, le but est de générer un texte qui maximise la probabilité d'appartenir à la classe cible. En raison de la taille de l'espace de toutes les séquences possibles ($|\mathcal{V}|^n$ pour une séquence de longueur n) ; il est infaisable d'évaluer $D(c \mid x)$ pour toutes les séquences possibles. Ainsi, les méthodes coopératives utilisent la distribution du modèle de langue génératif pour restreindre l'exploration aux séquences plausibles uniquement. De cette manière, la séquence produite maximise $p(x) * D(c \mid x) \propto p(x \mid c)$, résultant en une séquence qui est à la fois bien écrite et qui appartient à la classe cible.

3.2 Décodage guidé par *Monte Carlo Tree Seach*

Parmi ces approches coopératives, celles qui utilisent le *Monte Carlo Tree Search* (MCTS) pour guider le décodage des modèles de langue ont obtenu de très bons résultats (Scialom *et al.*, 2021a; Chaffin *et al.*, 2022; Leblond *et al.*, 2021; Lamprier *et al.*, 2022). Le MCTS est un algorithme itératif qui cherche une solution dans un arbre trop grand pour être parcouru de façon exhaustive. Il est adapté à la génération de texte car l'espace de recherche créé durant le décodage correspond à un arbre : la racine correspond à l'amorce et les enfants d'un nœud correspondent aux séquences construites en augmentant le préfixe de leur parent d'un token supplémentaire. La boucle du MCTS est composée de 4 étapes : sélection, expansion, simulation et rétro-propagation.

1. **Sélection.** Une exploration de la racine de l'arbre à une feuille non explorée. Le chemin vers la feuille est défini en sélectionnant, à chaque nœud, l'enfant qui maximise la *Polynomial Upper*

Confidence Trees (PUCT) (Rosin, 2011; Silver *et al.*, 2017) qui est définie, pour un nœud i par :

$$PUCT(i) = \frac{s_i}{n_i} + c_{puct}\ p(x_i \mid x_{1:t-1}) \frac{\sqrt{N_i}}{1+n_i} \tag{1}$$

avec n_i le nombre de simulations jouées après le nœud i, s_i le score agrégé du nœud, N_i le nombre de simulations jouées après son parent et c_{puct} une constante qui définit le compromis entre exploitation (se focaliser sur des nœuds avec des bons scores) et exploration (explorer des nœuds prometteurs).

2. **Expansion.** La création des enfants du nœud sélectionné s'il n'est pas terminal (i.e, correspondant au token de fin de séquence).
3. **Simulation (roll-out).** Le tirage de tokens additionnels (en utilisant la distribution du modèle de langue) jusqu'à un nœud terminal.
4. **Rétro-propagation.** L'évaluation de la séquence x associée au nœud terminal et l'agrégation de son score à tous les parents jusqu'à la racine. Pour guider la génération vers des textes qui appartiennent à une classe donnée, le score d'une séquence x associée à une feuille peut être défini par la probabilité $D(c \mid x)$ donnée par le classifieur. Différentes méthodes d'agrégation peuvent être utilisées, Chaffin *et al.* (2022) calcule la moyenne du score actuel et de celui du nœud terminal alors que (Scialom *et al.*, 2021b; Lamprier *et al.*, 2022) prennent le maximum des deux.

Cette boucle est répétée un certain nombre de fois, puis le token à ajouter à l'amorce est choisi grâce à l'arbre construit. Parmi les nœuds enfants de la racine, deux choix sont possibles : soit celui ayant été sélectionné le plus de fois, soit celui ayant un score agrégé le plus élevé. Comme nous souhaitons obtenir des séquences aussi stéréotypiques des classes du modèle discriminatif que possible, nous choisissons celui ayant le score le plus élevé. Ce nœud devient ensuite la nouvelle racine, et le processus est répété jusqu'à produire la séquence finale.

Contrairement aux méthodes traditionnelles qui décodent de gauche à droite et peuvent manquer des séquences qui deviennent meilleures après quelques étapes de génération ou se retrouver bloquer dans des séquences sous optimales, le MCTS brise le décodage myope en définissant le score d'un token sur la base des continuations possibles de la séquence. En plus d'être *plug-and-play*, c'est à dire que n'importe quel modèle de langue génératif (auto-régressif) peut être guidé durant le décodage par n'importe quel classifieur, cette approche a obtenu des résultats à l'état de l'art dans la tâche de génération contrainte, qui consiste à générer des textes qui maximisent $D(c \mid x)$ tout en maintenant une qualité d'écriture élevée.

4 Méthode

Dans cet article, nous présentons ***Therapy***, une méthode d'explication agnostique au modèle qui n'utilise pas de données en entrée. Therapy utilise un modèle de langue guidé par le classifieur à expliquer pour générer des textes représentatifs des classes apprises par le classifieur. Pour ce faire, Therapy extrait les mots importants pour le classifieur en l'utilisant pour guider le modèle de langue via la génération coopérative. Puisque les textes générés coopérativement suivent la distribution $p(x)D(c \mid x)$, leur distribution peut ensuite être utilisée pour étudier le classifieur D : les mots avec des fréquences élevées sont susceptibles d'être importants pour le classifieur. Ainsi, Therapy entraîne un modèle de régression logistique sur les représentations tf-idf des textes générés et retourne les

coefficients les plus importants et leurs termes associés comme explication. Comme $p(x)$ est le même pour toutes les classes, l'utilisation des tf-idf sur le corpus entier (i.e, les textes générés pour toutes les classes) filtre les mots qui sont fréquents uniquement à cause de $p(x)$ ou pour plusieurs classes. Les termes résultants sont donc dûs à la partie de la distribution venant du classifieur. L'entraînement d'un modèle de régression logistique sur les tf-idf permet d'extraire les termes les plus importants et d'étudier leur importance relative pour chaque classe. Therapy offre donc le niveau d'explicabilité d'une régression logistique basée sur des n-grams. Enfin, grâce à l'aspect plug-and-play de la génération guidée par MCTS, Therapy est une méthode agnostique au modèle qui peut expliquer tout type de modèle via n'importe quel modèle de langue et retourner une explication du fonctionnement global du classifieur.

En substance, la méthode est similaire à l'utilisation de LIME combinée à un modèle de langue qui remplace des tokens masqués lorsque le nombre de tokens remplacés tend vers l'infini mais avec deux avantages. Premièrement, la méthode ne repose pas sur des exemples en entrée mais génère ces exemples à partir de rien en utilisant le modèle de langue auto-régressif. Ceci est particulièrement utile pour les cas où les données ne peuvent pas être partagées pour des raisons de confidentialité (Amin-Nejad *et al.*, 2020). De plus, au lieu d'explorer le voisinage de ces exemples (et donc de conditionner les explications au contexte de ces exemples), le domaine d'exploration est défini par le modèle de langue. Ce modèle de langue peut être générique ou spécifique à un domaine sur lequel sera utilisé le classifieur pour s'assurer qu'il fonctionne correctement sur ce type de données précis.

Deuxièmement, la méthode ne génère pas **avant** de classifier mais utilise le classifieur **durant** la génération. Ainsi, au lieu de générer des textes "au hasard" et espérer que les caractéristiques importantes apparaissent, la méthode interroge explicitement le modèle pour des caractéristiques discriminantes via la maximisation de $D(c \mid x)$. Cela rend la méthode plus efficace et réduit la probabilité de générer (a) des mots rares mais qui ne sont pas importants pour le modèle, et (b) des textes "entre-deux" qui possèdent les caractéristiques de plusieurs classes et peuvent être perturbants. Par ailleurs, notre méthode s'appuie directement sur la distribution apprise par le classifieur étudié pour guider la génération, contrairement aux méthodes comme Polyjuice et GYC, qui en plus d'utiliser des exemples en entrée, s'appuient sur une distribution apprise par le modèle de langue pour biaiser la génération vers certaines caractéristiques (via les codes de contrôle).

Nous avons décidé d'appeler cette approche Therapy puisque nous associons son fonctionnement à celui d'un thérapeute (le LM). Ce thérapeute interroge son patient (le classifieur) afin de comprendre son comportement et éventuellement découvrir des comportements pathologiques (des biais).

5 Expériences

Dans cette section, nous présentons d'abord les détails expérimentaux de l'évaluation de Therapy (Section 5.1). Nous évaluons ensuite Therapy au travers de 3 expériences. La première expérience (Section 5.2), mesure la corrélation de Spearman des explications avec les poids de la boite transparente et étudie l'impact de la quantité de textes générés sur la qualité de l'explication retournée par le modèle linéaire. Nous comparons ensuite la capacité de Therapy à identifier les mots les plus importants pour le classifieur avec celles de LIME et SHAP dans la Section 5.3. Enfin, nous observons si les termes retournées par les différentes approches sont suffisants pour modifier la prédiction du classifieur. Le code de Therapy ainsi que celui des expériences sont disponibles publiquement.

5.1 Configuration expérimentale

Explication de boite transparente. Puisqu'il n'y a pas de vérité terrain disponible pouvant être utilisée comme objectif pour les méthodes d'explication évaluées, nous utilisons un modèle boite transparente. Un modèle est dit boite transparente lorsque ses paramètres utilisés pour faire une prédiction sont connus, on dit aussi que le modèle est explicable par conception. Tout au long de nos expérimentations, nous utilisons comme modèle boite transparente une régression logistique implémentée en utilisant sklearn (Pedregosa *et al.*, 2011). Les poids de ce modèle représentent le score d'importance associé à chacun des termes du vocabulaire.

Implémentation de Therapy. Lors de l'évaluation de la méthode proposée, nous utilisons l'implémentation de PPL-MCTS (Chaffin *et al.*, 2022) disponible sur GitHub. L'utilisation de la boite transparente dans PPL-MCTS se fait simplement en définissant la fonction qui prend en entrée une séquence et retourne son score. Le choix du modèle de langue génératif définit le domaine sur lequel nous voulons des explications pour le comportement du classifieur. Pour montrer que la méthode fonctionne bien avec un modèle de langue général (sans domaine particulier), nous utilisons OPT-125m (Zhang *et al.*, 2022). Une régression logistique est ensuite apprise sur la représentation tf-idf des textes générés et les coefficients de la régression logistique sont finalement retournés comme scores d'importance des différents tokens.

Jeu de données utilisés. Toutes les expériences ont été réalisées sur deux jeux de données différents issus de (Zhang *et al.*, 2015). Le premier, amazon_polarity : est un jeu de données de classification binaire composé de commentaires de produits Amazon labelisés comme positifs ou négatifs. Les textes qui le composent sont relativement courts et possèdent des champs lexicaux très caricaturaux. Le second, AG_news est un jeu de classification thématique à 4 classes (`world`, `sport`, `business` et `sci/tech`). Les textes de ce jeu sont plus longs et plus variés, mais ils possèdent différents indicateurs caractéristiques liés au fait qu'ils sont extraits d'articles de presse en ligne. Des exemples de textes générés par Therapy pour chacun des jeux de données ainsi que les premiers top-mots retournés sont disponibles en Annexe A.

Méthodes comparées. Nous comparons dans les Section 5.3 et 5.4, les résultats de Therapy avec les deux approches par attribution de caractéristiques les plus répandues : LIME (Ribeiro *et al.*, 2016) et SHAP (Lundberg & Lee, 2017) en utilisant les implémentations publiques. La différence principale entre LIME et SHAP est que la première génère un échantillon en perturbant une instance puis entraîne localement un modèle de régression linéaire tandis que la seconde utilise la théorie des jeux pour calculer l'importance de chaque élément. Nous avons utilisé les versions globales de ces méthodes sur 500 textes du jeu de test. Pour SHAP, nous avons décidé de garder les 10000 mots les plus importants pour chacun des jeux de données tandis que pour LIME, nous avons retourné les 500 explications locales avec les 35 mots les plus importants et regroupé les couples mots-valeurs dans un dictionnaire composé de 4592 mots pour amazon_polarity et 5770 pour ag_news.

5.2 Corrélation de Spearman

Une bonne explication de boite transparente est une liste de mots qui contient à la fois ses mots importants (i.e, a une bonne couverture) et les relie à des scores d'importance similaires. Ainsi, nous calculons la corrélation de Spearman entre les top mots de la boite transparente (ceux ayant un poids > 1) et leur score attribué par la méthode d'explication. La corrélation de Spearman à été préférée à celle de Pearson car les scores retournés par LIME et SHAP peuvent être très différents des poids de

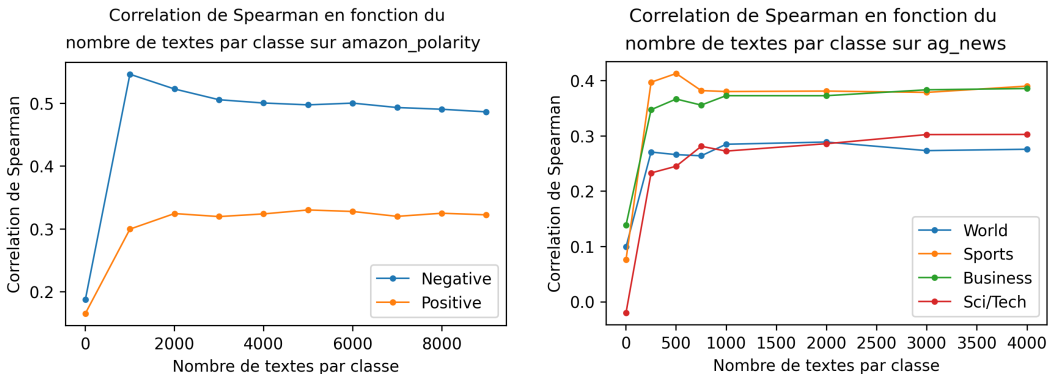


FIGURE 1 – Corrélation de Spearman en fonction du nombre de textes générés par classe pour amazon_polarity et ag_news

la boite transparente et une corrélation de rang donne donc une comparaison plus juste.

5.2.1 Influence du nombre de textes générés

Un des paramètres critiques de la méthode proposée est le nombre de textes à générer. En effet, un nombre élevé de tokens permet une plus grande couverture mais nécessite plus de calculs. Nous reportons la corrélation de Spearman en fonction du nombre de textes générées par classe dans la Figure 1. Nous pouvons voir que la corrélation augmente rapidement jusqu'à atteindre un plateau, ce qui signifie qu'une petite quantité de textes permet d'obtenir un bon aperçu du comportement du modèle et que la méthode ne nécessite pas énormément de calculs pour fonctionner. Pour la suite des expérimentations, nous fixons le nombre de textes générés par Therapy à 3000 par classe.

Jeu de données	AMAZON_POLARITY		AG_NEWS			
Classe	Positive	Negative	World	Sports	Business	Sci/Tech
LIME	0.64 (5.0e-7)	0.44 (1.5e-3)	0.09 (0.53)	0.16 (0.27)	0.20 (0.16)	0.19 (0.19)
LIME-other	0.21 (0.14)	0.18 (0.21)	-0.03 (0.85)	0.23 (0.12)	0.09 (0.52)	0.29 (0.04)
SHAP	0.71 (7.6e-9)	0.76 (1.6e-10)	0.47 (6.2e-4)	0.62 (1.7e-06)	0.53 (8.0e-5)	0.61 (2.4e-6)
SHAP-other	0.02 (0.87)	0.26 (0.06)	-0.05 (0.71)	0.04 (0.77)	0.15 (0.31)	0.12 (0.41)
Therapy	0.49 (3.3e-08)	0.31 (1.0e-4)	0.27 (1.6e-07)	0.37 (4.0e-12)	0.38 (5.6e-13)	0.3 (8.9e-09)

TABLE 1 – Corrélation de Spearman (p-valeur) entre les top mots de la boite transparente et les différentes méthodes d'explications. Les résultats sont donnés par classe et par jeu de données. Le suffixe 'other' indique que les explications sont générées à partir de l'autre jeu de données.

5.2.2 Comparaison avec les autres méthodes

Les corrélations de Spearman pour chacune des approches évaluées sont disponibles dans la Table 1. Les résultats obtenus par Therapy sont meilleurs que ceux de LIME sur ag_news mais moins bon sur amazon_polarity tandis que SHAP donne des résultats meilleurs que les autres méthodes sur les deux jeux de données. Ces résultats sont bons pour Therapy puisque LIME et SHAP génèrent des explications à partir du jeu de données de test, garantissant ainsi que les caractéristiques cibles se trouvent dans les exemples en entrée. Or, lorsque cette hypothèse ne tient plus, par exemple en

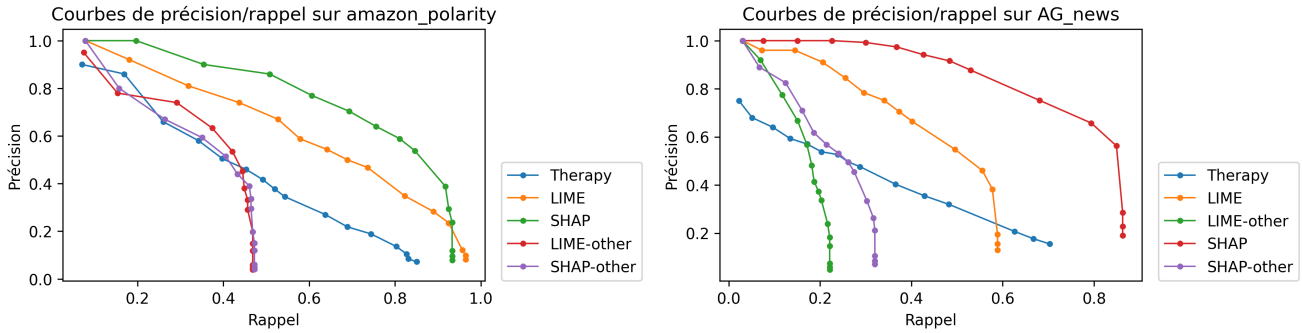


FIGURE 2 – Courbes de précision/rappel sur les top mots de la boite transparente (régression logistique) pour les différentes méthodes d'explications

utilisant comme données d'entrée le jeu de test de l'autre jeu de données (dénoté par *other* dans nos résultats), les méthodes ne parviennent plus à trouver les caractéristiques importantes et les corrélations chutent drastiquement, bien en dessous des résultats obtenus par Therapy.

5.3 Précision et rappel

Outre l'attribution correcte de scores aux caractéristiques importantes du modèle, il est également nécessaire que l'explication fournisse une sortie informative en pratique. Il faut donc s'assurer que les mots retournés dans l'explication (i.e, les mots avec les scores les plus élevés) soient effectivement des mots importants pour le modèle étudié et que ses mots les plus importants soient trouvés. Ainsi, la Figure 2 montre pour différents nombres de mots importants retournés, les valeurs de précision et de rappel moyennées sur toutes les classes. Le nombre k de mots retournés varie de 10 à 1500. La précision est obtenue en mesurant la proportion de mots retournés dans l'explication qui appartiennent au top mots de la boite transparente tandis que le rappel est la proportion des top mots de la boite transparente qui sont présents dans l'explication.

On observe que Therapy obtient de moins bons résultats que LIME (bien qu'obtenant un meilleur rappel sur ag_news) tandis que SHAP est meilleur que les deux méthodes sur les deux jeux de données. À nouveau, lorsque les données ne contiennent plus nécessairement les caractéristiques importantes pour le modèle (-other), les résultats s'écroulent et Therapy surpasse les deux approches. Cette limitation est visible par le plateau au niveau des scores de rappels pour ces méthodes : elles trouvent effectivement bien les caractéristiques importantes **présentes dans les données**, mais sont limitées à celles-ci uniquement, fixant la limite supérieure des caractéristiques pouvant être trouvées. En pratique, les biais contenus dans le modèle peuvent être suffisamment subtils pour ne pas être présents dans le jeu de données à disposition, auquel cas, LIME et SHAP ne peuvent pas les détecter. Therapy, au contraire, obtient de bons résultats en utilisant le même LM générique pour les deux jeux de données, sans utiliser d'apriori. La méthode permet donc d'obtenir un très bon aperçu du comportement du modèle lorsqu'aucune donnée, ou plus largement, lorsqu'aucune donnée représentative des caractéristiques importantes du modèle n'est disponible. Dans ce dernier cas de figure, Therapy permet d'offrir une recherche plus exhaustive que celles se basant sur des textes existants, obtenant un score de rappel important.

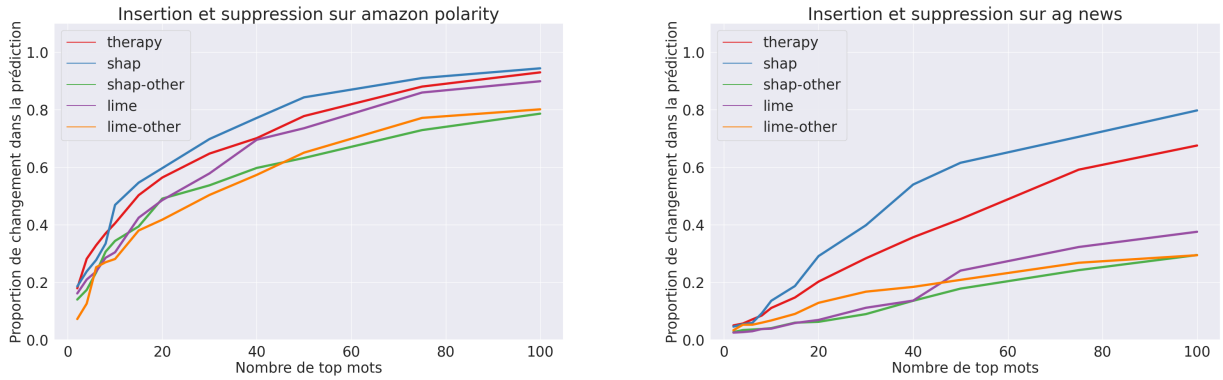


FIGURE 3 – Proportion de textes dont la classification change en fonction du nombre de mots importants utilisés pour effectuer le remplacement.

5.4 Insertion/suppression de mots importants

Une manière de valider l'exactitude des explications est de retirer les mots que l'explication indique comme étant importants pour une classe et d'observer comment les prédictions du modèle évoluent. L'intuition derrière cette approche est que supprimer la "cause" forcera le modèle à changer sa décision (Petsiuk *et al.*, 2018). De la même manière, rajouter un mot indiqué par l'explication comme étant important pour une autre classe devrait réduire la confiance du modèle. Ainsi, nous calculons une métrique d'insertion/suppression en mesurant la proportion de textes dont la classification par la boite transparente est modifiée lorsque l'on retire un mot étant indiqué comme important pour cette classe et que nous le remplaçons par un mot important pour une autre classe. La Figure 3 montre les résultats agrégés sur chacune des classes des deux jeux de données pour Therapy, LIME, SHAP et leurs versions utilisant l'autre jeu de données (-other). On observe que SHAP et Therapy sont les plus efficaces et obtiennent des résultats similaires, parvenant à modifier la prédiction du classifieur plus souvent que LIME à mesure que le nombre de mots échangés augmente. On remarque également que sur le jeu de données ag_news, où les mots importants sont moins communs que pour amazon_polarity, Therapy est capable de modifier la prédiction près de 70% du temps, tandis que SHAP et LIME n'arrivent à modifier la prédiction que dans 30% de cas en utilisant l'autre jeu de données. Ce qui montre encore une fois que ces méthodes nécessitent des données très spécifiques alors que Therapy est capable de trouver les mots importants pour chaque classe sans avoir accès à aucune donnée et sans utiliser d'apriori sur le modèle.

6 Conclusion

Les méthodes d'explicabilité usuelles s'appuient fortement sur des données en entrée, qui ne sont pas forcément disponibles et peuvent ne pas contenir les biais et caractéristiques importantes du modèle. Nous proposons Therapy, une méthode qui emploie la génération de texte coopérative afin de générer des données synthétiques qui suivent la distribution apprise par le modèle étudié. Ainsi, la recherche est dirigée par un modèle de langue génératif pré-entraîné et permet une exploration plus large que celle restreinte au voisinage des données d'entrée. Cela permet de relaxer la plupart des contraintes et aprioris induits par les méthodes qui se basent sur des exemples. Dans le cas extrême

où des données très représentatives (comme le jeu de test d'un jeu de données) des caractéristiques importantes du modèle sont disponibles, Therapy obtient des résultats légèrement moins bon que la méthode état-de-l'art SHAP, tout en restant compétitif avec LIME. Cependant, si on considère des cas d'usage plus réalistes dans lesquels les caractéristiques importantes ne sont pas explicitement données en entrée de la méthode d'explication, alors les performances de Therapy restent très bonnes et celles des autres méthodes s'effondrent (quand elles sont applicables). Ainsi, Therapy est un outil pertinent pour explorer le comportement d'un modèle lorsque la collecte des données sur lesquelles le modèle sera utilisée n'est pas possible ou très complexe.

Remerciements

Ces travaux sont financés par l'Agence Nationale de la Recherche (ANR) dans le cadre de la convention de subvention ANR-19-CE23-0019-01 et par le réseau TAILOR (EU Horizon 2020 programme d'innovation et de recherche avec la convention de subvention 952215).

Références

ABID A., FAROOQI M. & ZOU J. (2021). Persistent anti-muslim bias in large language models. In *AIES '21 : AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, p. 298–306 : ACM.

AMIN-NEJAD A., IVE J. & VELUPILLAI S. (2020). Exploring transformer text generation for medical dataset augmentation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 4699–4708, Marseille, France : European Language Resources Association.

ARASZKIEWICZ M., BENCH-CAPON T., FRANCESCONI E., LAURITSEN M. & ROTOLO A. (2022). Thirty years of artificial intelligence and law : overviews. *Artificial Intelligence and Law*.

ARRIETA A. B., RODRÍGUEZ N. D., SER J. D., BENNETOT A., TABIK S., BARBADO A., GARCÍA S., GIL-LOPEZ S., MOLINA D., BENJAMINS R., CHATILA R. & HERRERA F. (2020). Explainable artificial intelligence (XAI) : concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, **58**, 82–115.

BAKHTIN A., DENG Y., GROSS S., OTT M., RANZATO M. & SZLAM A. (2021). Residual energy-based models for text. *Journal of Machine Learning Research*, **22**(40), 1–41.

BODRIA F., GIANNOTTI F., GUIDOTTI R., NARETTO F., PEDRESCHI D. & RINZIVILLO S. (2021). Benchmarking and survey of explanation methods for black box models. *CoRR*.

BRAMHALL S., HORN H., TIEU M. & LOHIA N. (2020). QLIME-A : Quadratic Local Interpretable Model-Agnostic Explanation Approach. *SMU Data Science Rev*, **3**.

BROWN T. B., MANN B., RYDER N., SUBBIAH M., KAPLAN J., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., AGARWAL S., HERBERT-VOSS A., KRUEGER G., HENIGHAN T., CHILD R., RAMESH A., ZIEGLER D. M., WU J., WINTER C., HESSE C., CHEN M., SIGLER E., LITWIN M., GRAY S., CHESS B., CLARK J., BERNER C., MCCANDLISH S., RADFORD A., SUTSKEVER I. & AMODEI D. (2020). Language models are few-shot learners. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

BUCH V. H., AHMED I. & MARUTHAPPU M. (2018). Artificial intelligence in medicine : current trends and future possibilities. *Br. J. Gen. Pract.*, **68**(668), 143–144.

CHAFFIN A., CLAVEAU V. & KIJAK E. (2022). PPL-MCTS : constrained textual generation through discriminator-guided MCTS decoding. In M. CARPUAT, M. DE MARNEFFE & I. V. M. RUÍZ, Éds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, p. 2953–2967 : Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.215.

CHEN X., CAI P., JIN P., WANG H., DAI X. & CHEN J. (2020). Adding a filter based on the discriminator to improve unconditional text generation. *arXiv preprint arXiv :2004.02135*.

ELSHAWI R., SHERIF Y., AL-MALLAH M. & SAKR S. (2019). ILIME : Local and Global Interpretable Model-Agnostic Explainer of Black-Box Decision. In *ADBIS*.

ESTEVA A., KUPREL B., NOVOA R. A., KO J., SWETTER S. M., BLAU H. M. & THRUN S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, **542**(7639), 115–118.

GAUDEL R., GALÁRRAGA L., DELAUNAY J., ROZÉ L. & BHARGAVA V. (2022). s-LIME : Reconciling locality and fidelity in linear explanations. In *Advances in Intelligent Data Analysis XX - 20th International Symposium on Intelligent Data Analysis, IDA 2022, Rennes, France, April 20-22, 2022, Proceedings*, volume 13205 de *Lecture Notes in Computer Science*, p. 102–114 : Springer.

HOLTZMAN A., BUYS J., FORBES M., BOSSELUT A., GOLUB D. & CHOI Y. (2018). Learning to write with cooperative discriminators. In I. GUREVYCH & Y. MIYAO, Éds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1 : Long Papers*, p. 1638–1649 : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1152.

JACOVI A. (2023). Trends in explainable AI (XAI) literature. *CoRR*, **abs/2301.05433**. DOI : 10.48550/arXiv.2301.05433.

KARATZA P., DALAKLEIDI K., ATHANASIOU M. & NIKITA K. (2021). Interpretability methods of machine learning algorithms with applications in breast cancer diagnosis. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, p. 2310–2313. DOI : 10.1109/EMBC46164.2021.9630556.

LAMPRIER S., SCIALOM T., CHAFFIN A., CLAVEAU V., KIJAK E., STAIANO J. & PIWOWARSKI B. (2022). Generative cooperative networks for natural language generation. In K. CHAUDHURI, S. JEGELKA, L. SONG, C. SZEPESVÁRI, G. NIU & S. SABATO, Éds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 de *Proceedings of Machine Learning Research*, p. 11891–11905 : PMLR.

LEBLOND R., ALAYRAC J., SIFRE L., PISLAR M., LESPIAU J., ANTONOGLOU I., SIMONYAN K. & VINYALS O. (2021). Machine translation decoding beyond beam search. In M. MOENS, X. HUANG, L. SPECIA & S. W. YIH, Éds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, p. 8410–8434 : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.662.

LUCY L. & BAMMAN D. (2021). Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, p. 48–55, Virtual : Association for Computational Linguistics. DOI : 10.18653/v1/2021.nuse-1.5.

LUNDBERG S. M. & LEE S. (2017). A unified approach to interpreting model predictions. In I. GUYON, U. VON LUXBURG, S. BENGIO, H. M. WALLACH, R. FERGUS, S. V. N. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, p. 4765–4774.

MADAAN N., PADHI I., PANWAR N. & SAHA D. (2021). Generate your counterfactuals : Towards controlled counterfactual generation for text. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, p. 13516–13524 : AAAI Press.

MILLER T. (2019). Explanation in artificial intelligence : Insights from the social sciences. *Artif. Intell.*, **267**, 1–38. DOI : 10.1016/j.artint.2018.07.007.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M. & DUCHESNAY E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

PETSIUK V., DAS A. & SAENKO K. (2018). RISE : randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, p. 151 : BMVA Press.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVE I. (2018). Improving language understanding by generative pre-training.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

RIBEIRO M. T., SINGH S. & GUESTRIN C. (2016). "why should I trust you ?" : Explaining the predictions of any classifier. In B. KRISHNAPURAM, M. SHAH, A. J. SMOLA, C. C. AGGARWAL, D. SHEN & R. RASTOGI, Éds., *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, p. 1135–1144 : ACM. DOI : 10.1145/2939672.2939778.

ROBEER M., BEX F. & FEELDERS A. (2021). Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics : EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, p. 3611–3625 : Association for Computational Linguistics.

ROSIN C. D. (2011). Multi-armed bandits with episode context. *Ann. Math. Artif. Intell.*, **61**(3), 203–230. DOI : 10.1007/s10472-011-9258-6.

S. PUNLA C., HTTPS ://ORCID.ORG/ 0000-0002-1094-0018, CSPUNLA@BPSU.EDU.PH, C. FARRO R., HTTPS ://ORCID.ORG/0000-0002-3571-2716, RCFARRO@BPSU.EDU.PH & BATAAN PENINSULA STATE UNIVERSITY DINALUPIHAN, BATAAN, PHILIPPINES (2022). Are we there yet ? : An analysis of the competencies of BEED graduates of BPSU-DC. *International Multidisciplinary Research Journal*, **4**(3), 50–59.

SCIALOM T., DRAY P., LAMPRIER S., PIWOWARSKI B. & STAIANO J. (2020). Discriminative adversarial search for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 de *Proceedings of Machine Learning Research*, p. 8555–8564 : PMLR.

SCIALOM T., DRAY P., LAMPRIER S., PIWOWARSKI B. & STAIANO J. (2021a). To beam or not to beam : That is a question of cooperation for language gans. *Advances in neural information processing systems*.

SCIALOM T., DRAY P., STAIANO J., LAMPRIER S. & PIWOWARSKI B. (2021b). To beam or not to beam : That is a question of cooperation for language gans. In M. RANZATO, A. BEYGELZIMER, Y. N. DAUPHIN, P. LIANG & J. W. VAUGHAN, Éds., *Advances in Neural Information Processing Systems 34 : Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, p. 26585–26597.

SHANKARANARAYANA S. M. & RUNJE D. (2019). ALIME : Autoencoder Based Approach for Local Interpretability. *CoRR*, **abs/1909.02437**.

SILVER D., SCHRITTWIESER J., SIMONYAN K., ANTONOGLOU I., HUANG A., GUEZ A., HUBERT T., BAKER L., LAI M., BOLTON A., CHEN Y., LILLICRAP T. P., HUI F., SIFRE L., VAN DEN DRIESSCHE G., GRAEPEL T. & HASSABIS D. (2017). Mastering the game of go without human knowledge. *Nat.*, **550**(7676), 354–359. DOI : 10.1038/nature24270.

TAGARELLI A. & SIMERI A. (2022). Unsupervised law article mining based on deep pre-trained language representation models with application to the italian civil code. *Artificial Intelligence and Law*, **30**(3), 417–473.

VISANI G., BAGLI E. & CHESANI F. (2020). Optilime : Optimized LIME explanations for diagnostic computer algorithms. In *Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Galway, Ireland, October 19-23, 2020*, volume 2699 de *CEUR Workshop Proceedings* : CEUR-WS.org.

WU T., RIBEIRO M. T., HEER J. & WELD D. S. (2021). Polyjuice : Generating counterfactuals for explaining, evaluating, and improving models. In C. ZONG, F. XIA, W. LI & R. NAVIGLI, Éds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1 : Long Papers), Virtual Event, August 1-6, 2021*, p. 6707–6723 : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.523.

ZAFAR M. R. & KHAN N. M. (2019). DLIME : A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems. *CoRR*, **abs/1906.10263**.

ZHANG S., ROLLER S., GOYAL N., ARTETXE M., CHEN M., CHEN S., DEWAN C., DIAB M. T., LI X., LIN X. V., MIHAYLOV T., OTT M., SHLEIFER S., SHUSTER K., SIMIG D., KOURA P. S., SRIDHAR A., WANG T. & ZETTLEMOYER L. (2022). OPT : open pre-trained transformer language models. *CoRR*, **abs/2205.01068**. DOI : 10.48550/arXiv.2205.01068.

ZHANG X., ZHAO J. J. & LECUN Y. (2015). Character-level convolutional networks for text classification. In C. CORTES, N. D. LAWRENCE, D. D. LEE, M. SUGIYAMA & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 28 : Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, p. 649–657.

A Résultats qualitatifs

Nous donnons dans cette annexe, quelques exemples d'échantillons générés par Therapy pour les différentes classes des deux jeux de données étudiés ainsi que les 20 premiers top-mots retournés par la méthode. Veuillez noter que certains "mots" correspondent à des "sous-mots", dû à la décomposition en unigrammes (ve, ll, ...). L'approche proposée permet d'étudier l'impact de n-grammes, mais ceci n'étant pas possible avec LIME et SHAP (avec le code disponible), nous avons limité l'étude aux unigrammes.

A.1 amazon_polarity, classe "positive"

Échantillons :

— "The amount of support they gave was amazing. I'm so happy with the game and it's a great way for me to play my favorite role! Also, I love that they're giving us a new character every week, which is awesome :)I think they've been pretty good lately too. It's nice when we see someone who really loves their team and has fun doing things"

— "The whole point of this post is to show people what a great job you've put in. You are doing an excellent job, I love this sub. Keep going. :)I'm so happy for all those people that have helped! It's amazing to be here and it was fun being the one that got the most attention on the other day :). Thank you so much <3You're very welcome :) And yes... I am glad you enjoyed this subreddit - it is a great place"

Top-mots : *great, love, good, ve, years, people, lot, friends, fun, life, world, works, easy, things, happy, heard, including, awesome, nice, family*

A.2 amazon_polarity, classe "negative"

Échantillons :

— "I guess you could say it has to be the most boring thing ever.It is a very boring experienceI don't think they have any idea how to make money off this, so they will probably just do what you say they will if it doesn't work. They may try to sell you on the fact that it isn't working and then try and get you to buy back into it when they are done making money. If I was them, would they not want to pay me?"

— "It might be a bug, but it doesn't work on my phoneYeah that sucks! I was hoping the same thing happened.I don't think you can get a refund without having to buy a new one...You have no idea what they would be like without the warranty. They wouldn't know how much money is left if they didn't do it

properly or at all and they would probably just give it back as soon as they found out it was defective (or something)"

Top-mots : *don, money, bad, doesn, didn, idea, work, device, isn, thing, guess, wrong, back, buy, fact, time, phone, point, problem, thought*

A.3 ag_news, classe "world"

Échantillons :

— The world's most popular and sought after luxury home in Dubai, the Al Khaleel International Airport (ALI), is a city that is renowned as the hub of the Middle East and North Africa region. It is also the capital city of Abu Dhabi, United Arab Emirates with a population over 2 billion people and a GDP per person.Al-Khaliel airport is an important gateway into Europe to the north east and west of Europe, Asia, and Australia

— In the last month, the US government has been accused of using a "secret" military intelligence agency in Iraq and Afghanistan for spying on its allies, including Pakistan, Turkey, Iran, Russia, Syria, Sudan, Lebanon, Libya, Egypt, Nigeria...and even Saudi Arabia!According to a new report published today, the Pentagon is now investigating the alleged use by the US military Intelligence Community (IUC), which was created by President Bush after the 9/11 terror attack that killed over

Top-mots : *people, man, country, city, party, killed, family, agree, wrong, general, children, sex, president, police, working, military, dead, missing, woman, days*

A.4 ag_news, classe "sport"

— I think it's pretty safe to assume that the guy has a lot of experience with the game. He played in the NHL, and he was a top 10 player on the team for most games last season (he had a goal in his final three playoff series), and he won a Stanley Cup as a rookie this past season (he finished third in the league in points scored, which was good for second in the league) and is still one of the best players in hockey at this stage in the year

— We've got to keep playing this game. This team needs to win games and we need to play the best basketball that's been in our league all season, every night. And it is time to get out there and do that.""The Warriors have won three straight games at home and are 2-0 against the NBA's best team on both sides of the ball this season with a record of 21-1 (13.7 points per game)

Top-mots : *time, game, back, season, play, didn, team, guy, field, night, games, left, 12, title, won, saturday, playing, great, day, wasn*

A.5 ag_news, classe "business"

Échantillons :

— I am still in shock after hearing of that.It's a pretty big deal. It happened last month. They are trying to get the money out of the company by selling their stock for profit so they can sell more shares and buy more shares at higher prices (which I think would have helped with the stock market) and it was reported as an ""investment fraud"" by the SEC which has been going on all over this subreddit for months, but no one ever seems to care much

— Biden is planning to spend millions of dollars to buy a new home, but the real estate market in America is still struggling with the housing shortage. The average house sale cost $1 billion and was up by nearly 50 percent from the previous year's price of about $800 million -- according to the Real Estate Board of New York (RBE).The RBE estimates that the average house sales prices are expected to rise 1,000 per month this fiscal year as the economy continues its rebound

Top-mots : *money, buy, care, doesn, things, deal, pay, worth, business, car, biggest, interested, month, trade, don, compagny, happened, store, kind, price*

A.6 ag_news, classe "sci/tech"

Échantillons :

— 2K Games' Dark Souls 3 is coming to PC, Mac & Linux in the near future.The new game will launch for free on PC, Mac & Linux and Xbox One, PlayStation 5 and Microsoft Windows, as well. It'll come out sometime during this week, with an official release expected soon thereafter, though we don't yet know what it will be called or where exactly you're getting the title. We also have some news from Sony that's not quite so surprisingetc...

— In this new age of technology, the world needs more people. We have a lot in our hands. The internet can help us connect to others through video chat and online games.""The company will launch a mobile game called 'Gangster', where it plans to offer ""an interactive experience"" with its users, according to the company. The game has been developed for the Apple iPad and Android phones that use Apple TV, which also uses Google Chromecast, according to a release.

Top-mots : *ve, ll, idea, phone, internet, make, system, video, online, life, understand, version, pc, found, 13, thing, computer, lot, hard, issue, people, work, information, future*

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Extraction de relations sémantiques et modèles de langue : pour une relation à double sens

Olivier Ferret
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
olivier.ferret@cea.fr

RÉSUMÉ

Les modèles de langue contextuels se sont rapidement imposés comme des outils essentiels du Traitement Automatique des Langues. Néanmoins, certains travaux ont montré que leurs capacités en termes de sémantique lexicale ne les distinguent pas vraiment sur ce plan de modèles plus anciens, comme les modèles statiques ou les modèles à base de comptes. Une des façons d'améliorer ces capacités est d'injecter dans les modèles contextuels des connaissances sémantiques. Dans cet article, nous proposons une méthode pour réaliser cette injection en nous appuyant sur des connaissances extraites automatiquement. Par ailleurs, nous proposons d'extraire de telles connaissances par deux voies différentes, l'une s'appuyant sur un modèle de langue statique, l'autre sur un modèle contextuel. Des évaluations réalisées pour l'anglais et focalisées sur la similarité sémantique ont montré l'intérêt de cette démarche, permettant d'enrichir sémantiquement un modèle de type BERT sans utilisation de ressources sémantiques externes.

ABSTRACT

Extraction of semantic relations and language models : for a two-way relationship

Contextual language models have rapidly become essential tools in Natural Language Processing. Nevertheless, some works have shown that their capabilities in terms of lexical semantics do not really distinguish them from older models, such as static models or count-based models. One way to improve these capabilities is to inject semantic knowledge into contextual models. In this paper, we propose a method to perform this injection based on automatically extracted knowledge. Moreover, we propose to extract such knowledge in two different ways, one based on a static language model, the other on a contextual model. Evaluations performed for English and focused on semantic similarity have shown the interest of this approach, allowing to semantically enrich a BERT model without using external semantic resources.

MOTS-CLÉS : Extraction de relations sémantiques lexicales, modèles de langue, injection de connaissances dans les modèles de langue.

KEYWORDS: Extraction of lexical semantic relations, language models, knowledge injection in language models.

1 Introduction

Les modèles de langue, qu'ils soient à base de comptes ou prédictifs (Baroni *et al.*, 2014), et parmi ces derniers, statiques ou contextuels (Naseem *et al.*, 2021), entretiennent une relation double vis-à-vis des connaissances sémantiques. D'une part, du fait de leur forte inscription dans l'hypothèse

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 123–136.

distributionnelle (Harris, 1954), ils constituent un moyen utilisé de longue date pour extraire des relations sémantiques lexicales à partir de corpus (Lenci *et al.*, 2022). D'autre part, beaucoup de travaux se sont attachés au problème de l'injection de connaissances sémantiques dans ces modèles afin de les enrichir (Wang *et al.*, 2023), soit dans une perspective générale d'amélioration de la prise en compte des phénomènes sémantiques au niveau des tâches auxquels ils sont appliqués, soit pour leur adaptation à des domaines spécifiques.

L'utilisation des modèles de langue pour l'extraction de relations sémantiques est étroitement liée à la problématique de la similarité sémantique (Budanitsky & Hirst, 2006) et à celle des thésaurus distributionnels (Grefenstette, 1994; Lin, 1998; Curran & Moens, 2002). La façon la plus commune d'extraire des relations sémantiques à partir d'un modèle de langue est en effet de s'appuyer sur la capacité de ces modèles à évaluer la similarité des mots les uns par rapport aux autres sur une base distributionnelle, capacité utilisée par ailleurs pour l'évaluation intrinsèque de ces modèles (Faruqui *et al.*, 2016). Appliquée au vocabulaire d'un corpus, cette capacité permet de construire un thésaurus distributionnel donnant pour chaque mot cible une liste de voisins distributionnels, ordonnés selon la valeur décroissante de leur similarité, évaluée par un modèle de langue, avec le mot cible. Les premiers voisins sont alors supposés les plus pertinents sur le plan sémantique, avec un biais de principe vers les relations paradigmatiques compte tenu de l'hypothèse distributionnelle sous-jacente aux modèles de langue. Compte tenu de ce principe général, la voie principale d'amélioration de cette extraction concerne la similarité sémantique utilisée pour construire les thésaurus distributionnels (Padró *et al.*, 2014a,b). Néanmoins, quelques travaux se concentrent également sur une amélioration des thésaurus en tant que tels au moyen de méthodes de réordonnancement, soit à un niveau global (Claveau *et al.*, 2014), soit plus localement au niveau de chaque entrée du thésaurus (Ferret, 2013a).

La question de l'injection de connaissances sémantiques dans les modèles de langue a fait quant à elle l'objet d'un grand nombre d'études, d'abord axées sur les modèles neuronaux statiques pour ensuite se focaliser sur les modèles contextuels. Malgré les différences existant entre ces deux grands types de modèles, ils partagent la même distinction entre les méthodes opérant lors de la construction du modèle et celles venant enrichir un modèle après sa construction. Les secondes ont clairement l'avantage du nombre dans le cas des modèles statiques, dans le prolongement de Faruqui *et al.* (2015), tandis que la situation est plus contrastée pour les modèles contextuels. En se limitant aux relations sémantiques lexicales [1], on peut ainsi citer le modèle LIBERT de Lauscher *et al.* (2020) pour la première catégorie de méthodes et le modèle LexFit de Vulić *et al.* (2021) pour la seconde.

Le travail présenté dans cet article conjugue les deux dimensions esquissées ci-dessus : il enrichit un modèle neuronal contextuel de type BERT (Devlin *et al.*, 2019) par l'injection de connaissances sémantiques lexicales mais contrairement aux travaux existants, ces connaissances sont elles-mêmes extraites automatiquement par le biais de l'exploitation de modèles de langue neuronaux. Plus précisément, les contributions de ce travail sont :

— la proposition et l'évaluation d'une nouvelle méthode d'extraction de relations sémantiques lexicales entre termes simples en appliquant une tâche de type « mot masqué » à des termes complexes par le biais d'un modèle contextuel ;
— la comparaison et l'association des relations ainsi obtenues avec les relations extraites à partir d'un modèle de langue statique ;
— l'évaluation de l'intérêt de l'utilisation de relations lexicales sémantiques extraites automatiquement pour enrichir un modèle de langue contextuel.

1. Pour les modèles contextuels, les travaux existants sont axés sur des graphes de connaissances représentant des connaissances factuelles plus que sur des relations sémantiques lexicales, la tendance étant inverse pour les modèles statiques.

2 Méthodes

Dans ce qui suit, nous présentons d'abord à la section 2.1 deux méthodes d'extraction de relations sémantiques lexicales à partir de modèles neuronaux de types différents. Dans les deux cas, les relations extraites caractérisent une relation de similarité sémantique entre deux mots mais ne sont pas typées. L'union du produit de chacune de ces deux méthodes sert ensuite de base pour l'injection de connaissances sémantiques dans un modèle de type BERT, objet de la section 2.2.

2.1 Extraction de relations sémantiques

À partir d'un modèle neuronal statique. Pour extraire un premier ensemble de relations de similarité sémantique, nous transposons à un modèle neuronal statique le principe de sélection par réciprocité dans le graphe des k plus proches voisins (*k-NN*) présenté dans (Claveau *et al.*, 2014) pour des modèles à base de comptes. Plus précisément, pour chaque mot cible, ses k plus proches mots voisins sont extraits en s'appuyant sur les similarités données par les plongements du modèle statique considéré, en l'occurrence un modèle Skip-gram (Mikolov *et al.*, 2013a). Cette extraction est réalisée grâce à la bibliothèque Faiss (Johnson *et al.*, 2021) en utilisant classiquement la mesure de similarité *cosinus* [2]. La relation de voisinage distributionnel n'est pas symétrique par nature mais nous utilisons précisément l'observation d'une telle symétrie comme critère de sélection des relations de voisinage les plus représentatives en termes de similarité sémantique. Plus précisément, une telle relation entre les mots x et y est sélectionnée si y se trouve parmi les k premiers voisins distributionnels de x et réciproquement, si x se trouve parmi les k premiers voisins distributionnels de y.

À partir d'un modèle neuronal contextuel. La transposition de l'approche précédente des modèles neuronaux statiques aux modèles neuronaux contextuels est bien moins directe que celle des modèles à base de comptes aux modèles neuronaux statiques, en particulier parce qu'un modèle contextuel produit par définition des représentations de mots en contexte et non des représentations génériques. Le problème plus généralement posé pour réaliser cette transposition est de pouvoir construire à partir d'un modèle de langue un graphe de voisinage entre mots, le voisinage étant fondé sur la notion de similarité sémantique. Pour un modèle contextuel, deux stratégies principales sont envisageables :

— la construction de plongements statiques de mots, ce qui permet de se ramener à la configuration évoquée au point précédent ;
— l'exploitation des capacités d'un tel modèle pour la tâche de modélisation du langage à partir de laquelle il a été entraîné.

La première stratégie a déjà fait l'objet d'un certain nombre de travaux (Ethayarajh, 2019; Bommasani *et al.*, 2020; Vulić *et al.*, 2020; Ferret, 2022), avec deux variantes principales : l'une considère pour un mot cible un ensemble de phrases contenant ce mot et agrège, généralement par une moyenne, les représentations contextuelles produites par le modèle de langue pour ce mot dans chacune de ces phrases [3]. La seconde variante consiste à construire une représentation à partir d'une seule occurrence du mot cible en isolation, sans le contexte d'une phrase. Néanmoins, Ferret (2022) montre que du point de vue de la constitution d'un voisinage sémantique des mots, cette première stratégie ne donne pas de résultats notablement plus intéressants que des plongements statiques, avec tout de même un

2. Concrètement, nous utilisons l'index IndexFlatIP, conçu pour les recherches exactes fondées sur le produit scalaire.
3. Les modèles de langue contextuels existants étant constitués de plusieurs couches, la représentation d'une occurrence de mot admet elle-même différentes variantes.

avantage à la première variante par rapport à la seconde.

Nous avons donc opté pour la seconde stratégie. Nous nous concentrons ici sur les modèles de type BERT, qui reposent sur une tâche de modélisation du langage par mot masqué (*Masked Language Modeling*). Néanmoins, l'approche n'exclut pas pour autant l'utilisation de modèles auto-régressifs de type GPT (Radford *et al.*, 2018). Le principe général s'inspire de l'utilisation de modèles de langue de type BERT pour la substitution lexicale sans utilisation de substituts de référence (Zhou *et al.*, 2019). Néanmoins, au lieu de considérer des occurrences de mots dans le contexte de phrases, nous nous limitons à des occurrences de mots au sein de termes complexes. L'application de la substitution lexicale aux termes complexes se retrouve par ailleurs dans (Wang, 2022) mais dans le cadre de phrases et avec l'objectif différent de valider des relations sémantiques entre termes complexes à partir de relations connues entre termes simples. Dans notre cas, la restriction aux termes complexes, plus spécifiquement de nature nominale, se justifie en premier lieu par des raisons de coût de calcul, le traitement de termes par un modèle de type BERT étant nettement moins coûteux que celui de phrases [4]. Par ailleurs, les expériences concernant la similarité distributionnelle avec les modèles à base de compte ou les modèles neuronaux statiques montrent de façon récurrente que la similarité sémantique, par opposition à la proximité sémantique, est mieux capturée par un contexte étroit que par un contexte large, ce qui justifie de se limiter à des termes complexes. L'analyse des schémas d'attention dans les modèles BERT (Clark *et al.*, 2019) montre en outre que certaines de leurs têtes prennent en compte spécifiquement ces interactions à courte portée, laissant à penser que ce choix n'est pas trop limitant. Enfin, cette méthode permet également, dans le cadre de domaines de spécialité, d'exploiter non seulement des termes extraits de corpus mais également des termes issus de terminologies de référence pour ces domaines.

Concrètement, l'approche consiste à soumettre à un modèle BERT en mode prédiction de mot masqué un ensemble de termes dont l'un des constituants a été masqué et de recueillir les k premières prédictions du modèle, avec leur score, en excluant le constituant à prédire. Chaque terme ainsi soumis constitue une séquence autonome. L'hypothèse est que les prédictions obtenues correspondent à des voisins sémantiques du constituant cible. Appliqué à un grand nombre de termes complexes, cette méthode conduit à recueillir des voisins sémantiques pour un ensemble conséquent de mots simples, ce qui permet ensuite d'appliquer le principe de sélection par réciprocité dans le graphe des *k-NN* vu ci-dessus. Un point important de l'approche est le fait qu'un même mot peut se voir associer autant de listes de voisins que le nombre de fois où il apparaît comme mot masqué dans un terme complexe. Pour constituer une liste de voisins unique pour chaque mot, nous appliquons une méthode de fusion de listes, en l'occurrence la méthode CombSum (Fox & Shaw, 1994), en exploitant les scores de prédiction normalisés avec la méthode Zero-one (Lee, 1997; Wu *et al.*, 2006). Outre le fait d'assurer l'unicité de la liste des voisins d'un mot, cette fusion a l'intérêt de mettre en avant les substituts prédits régulièrement avec les meilleurs scores et donc de placer aux premiers rangs les voisins supposés les plus proches du mot cible.

La prédiction d'un modèle BERT concernant un mot à substituer est clairement dépendante du contexte linguistique de ce mot. Dans notre cas, ce contexte est déterminé par plusieurs facteurs : la forme des termes complexes dans lesquels ces mots apparaissent, le rôle que les mots cibles y jouent et enfin, le contexte plus général dans lequel les termes complexes sont placés. Pour ce qui est du premier facteur, le travail présenté ayant été réalisé pour l'anglais avec des noms pour cibles, nous avons observé, en prenant comme base la version anglaise de Wikipédia, les termes composés

4. La complexité du mécanisme d'attention des transformeurs est quadratique en fonction de la longueur de la séquence considérée.

incluant deux mots pleins, obtenant ainsi les trois structures de termes suivantes, la première étant environ deux fois plus fréquente que la deuxième, qui est elle-même environ vingt fois plus fréquente que la troisième [5] :

ADJ	NOM		rough estimate, wearable device, motherless child
NOM	NOM		prison guard, science academy, college student
NOM	PREP	NOM	lack of food, degree in education, return on investment

Concernant le dernier facteur, nous avons repris le schéma général proposé par Qiang *et al.* (2020) dans un contexte de simplification lexical et consistant à conditionner la structure contenant une unité à prédire par cette même structure sous sa forme complète. Dans notre cas de figure, si TERM désigne le terme utilisé comme contexte immédiat et TERM_MSK, ce même terme avec le « trou » correspondant au mot cible, la séquence, appelée amorce, soumise à un modèle de type BERT en mode prédiction de mot masqué est ainsi de la forme :

	TERM . [SEP] TERM_MSK .
soit par exemple ici :	ADJ NOM . [SEP] ADJ __ .
qui peut s'instancier en :	civil defence . [SEP] civil __ .
	black magic . [SEP] black __ .

où __ correspond à l'emplacement du mot cible masqué et [SEP] au marqueur de changement de séquence [6]. Cette amorce est appelée P0 dans ce qui suit.

Nous avons également testé les variantes suivantes, destinées en particulier à donner un contexte immédiat un peu plus large :

P1 this is a/an TERM . [SEP] this is a/an TERM_MSK .
P2 TERM . [SEP] this is a/an TERM_MSK .
P3 a/an TERM . [SEP] a/an TERM_MSK .
P4 TERM . [SEP] a/an TERM_MSK is a kind of TERM .
P5 TERM . [SEP] a/an TERM_MSK is a/an TERM .
P6 TERM . [SEP] a/an TERM is a/an TERM_MSK .
P7 TERM . [SEP] a/an TERM_MSK and a/an TERM .
P8 TERM . [SEP] a/an TERM_MSK or a/an TERM .

2.2 Injection de relations sémantiques dans un modèle contextuel

Pour l'injection des relations sémantiques extraites, nous nous sommes appuyés sur une approche contrastive relevant de l'apprentissage de métriques. Ce type d'approches a déjà été exploré pour réaliser cette tâche d'injection aussi bien pour les plongements statiques (Shah *et al.*, 2020) que pour les modèles de langue contextuels (Vulić *et al.*, 2021). Dans le cas présent, nous nous situons dans le cadre défini par Vulić *et al.* (2021), qui ont eux-mêmes réutilisé le cadre défini par Sentence-BERT pour la similarité de phrases (Reimers & Gurevych, 2019). Plus précisément, l'architecture de base

5. ADJ : adjectif ; PREP : préposition.
6. Nous avons repris l'utilisation de [SEP] de (Qiang *et al.*, 2020) mais la nécessité de sa présence reste à tester, en particulier dans le cas de l'utilisation d'un modèle de type RoBERTa (Liu *et al.*, 2019), qui n'est pas entraîné pour la tâche de prédiction de la phrase suivante.

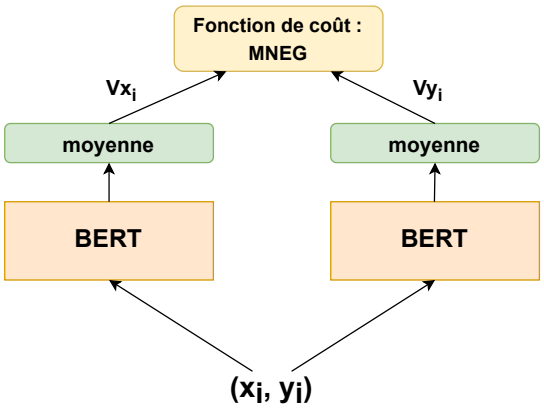


FIGURE 1: Architecture du modèle d'injection de relations sémantiques lexicales

de Sentence-BERT est celle d'un réseau siamois exploitant un encodeur double (*dual encoder*) : deux phrases dont on connaît la similarité sont encodées séparément par le même modèle de type BERT, une représentation de chacune des deux phrases est construite via un processus de regroupement (*pooling*) et les deux représentations ainsi obtenues sont prises en compte par une fonction de coût visant à rapprocher, au travers du mécanisme de rétropropagation, les représentations des phrases connues comme similaires tandis qu'elles tendent à écarter les représentations des phrases connues comme dissemblables. Le modèle LexFit de Vulić *et al.* (2021) réutilise directement cette architecture en donnant en entrée des couples de mots plutôt que des couples de phrases et en utilisant des relations lexicales sémantiques comme référence en termes de similarité, comme illustré par la figure 1. De nombreuses fonctions de coût sont possibles pour mettre en œuvre le principe général ci-dessus, et donc indirectement injecter ces relations lexicales dans un modèle de langue de type BERT ; mais au vu des expérimentations faites avec LexFit, nous avons choisi la Multiple Negatives Ranking (MNEG) loss (Henderson *et al.*, 2019), qui se définit ainsi pour un lot (*batch*) de B paires de mots $(x_1, y_1), \ldots, (x_B, y_B)$ telles que chaque paire (x_i, y_i) est sous-tendue par une relation sémantique :

$$\mathcal{L} = -\sum_{i=1}^{B} S(x_i, y_i) + \sum_{i=1}^{B} \log \sum_{j=1, j\neq i}^{B} e^{S(x_i, y_j)} \quad (1)$$

Cette fonction de coût permet d'adapter le modèle de langue de l'encodeur de façon à maximiser la similarité de chaque paire de mots (x_i, y_i) du lot (1^{er} terme de l'équation 1) tout en minimisant la similarité des $B - 1$ paires (x_i, y_j) (2^{nd} terme de l'équation 1) formées chacune d'un x_i et de l'y_j de toutes les autres paires du lot, ces (x_i, y_j) étant considérées comme des exemples négatifs de similarité. $S(x_i, y_i)$ correspond à la fonction utilisée pour évaluer la similarité de la paire (x_i, y_i).

3 Expérimentations

3.1 Cadre d'évaluation

Pour évaluer les résultats de notre processus d'injection de relations sémantiques lexicales dans un modèle de langue, nous avons choisi de tester les capacités du modèle cible en termes de similarité sémantique par le biais de plongements construits à partir de ce modèle. À la suite de Budanitsky &

Hirst (2006), nous distinguons la similarité sémantique, incarnée par les relations paradigmatiques (synonymie, hyperonymie...), de la proximité sémantique, que l'on retrouve dans les relations sémantiques de nature plus syntagmatique (comme les relations de prédication par exemple). Notre modèle cible est un modèle de type BERT, dans sa version `base-uncased`, et à l'instar de (Vulić *et al.*, 2021), nous construisons le plongement d'un mot à partir de l'encodage par ce modèle d'une seule occurrence de ce mot hors contexte en sélectionnant la représentation de cette occurrence produite au niveau de l'une des 13 couches du modèle (12 couches internes plus la couche d'entrée). Par ailleurs, comme (Bommasani *et al.*, 2020), lorsqu'un mot se décompose en plusieurs sous-mots (*wordpieces*), nous construisons sa représentation en moyennant les représentations de ses sous-mots.

L'évaluation en elle-même s'appuie sur la similarité entre les représentations ainsi produites pour les mots : pour chaque mot cible w_i, l'ensemble de ses k plus proches voisins est sélectionné en calculant la similarité de w_i avec tous les autres mots cibles w_j, en appliquant la mesure *cosinus* à leurs représentations et en ordonnant ces mots selon la valeur décroissante de leur similarité avec w_i. En pratique, $k = 10$ et le calcul des similarités est réalisée là encore grâce à la bibliothèque Faiss. Nous évaluons l'exactitude de ce classement comme en recherche d'information grâce à la R-précision ($R_{préc.}$), la MAP (Mean Average Precision) et aux précisions à différents rangs (P@r). Les évaluations ont été menées pour 10 305 noms cibles déjà utilisés dans (Ferret, 2022) et couvrant un large éventail de fréquences.

Tout en nous concentrant globalement sur les relations de nature paradigmatique, nous considérons deux références, toutes deux issues de WordNet (Miller, 1990) puisque nos mots cibles sont des noms communs en anglais : *para*, qui rassemble les relations de synonymie, d'hyponymie, d'hyperonymie et de cohyponymie et *syn*, qui se limite au sous-ensemble des synonymes. Nous définissons plus précisément les mots liés à un mot cible par ces différentes relations de la façon suivante :

— synonymes : tous les mots faisant partie d'un synset S_i du mot cible considéré ;
— hyperonymes : tous les mots des synsets S_{hype} ayant un lien direct d'hyperonymie avec un S_i ;
— hyponymes : tous les mots des synsets ayant une relation directe d'hyponymie avec S_i ;
— cohyponymes : tous les mots des synsets, à l'exception des S_i, ayant une relation directe d'hyponymie avec les synsets S_{hype}.

3.2 Mise en œuvre des méthodes proposées

La mise en œuvre de la première méthode d'extraction de relations lexicales sémantiques nécessite de disposer de plongements statiques tandis que la seconde demande principalement un ensemble de termes complexes. Pour ces deux ressources, nous nous sommes appuyés sur un même corpus de base, en l'occurrence un dump de Wikipédia en anglais du 1/10/2018 comprenant 2,16 milliards de tokens [7], étiquetés et lemmatisés grâce à l'outil CoreNLP (Manning *et al.*, 2014), dans sa version 3.9.2. Les termes complexes, limités ici à des bigrammes de mots pleins, ont été extraits par la méthode définie dans (Mikolov *et al.*, 2013b) [8], avec une fréquence minimale des termes extraits égale à 5 et un seuil d'information mutuelle minimale égal à 0. Nous avons ainsi obtenu 394 024 termes de structure ADJ NOUN pour constituer ensuite nos amorces. Pour chaque amorce, nous ne retenons que les 10 premières propositions du modèle, hors le terme masqué si celui-ci apparaît dans les premières propositions. Les plongements statiques ont été appris suivant le modèle Skip-gram avec l'outil *word2vec* [9] à partir de la forme lemmatisée des mots.

7. https://www.dropbox.com/s/cnrhd11zdtc1pic/enwiki-20181001-corpus.xml.bz2?dl=0
8. Suivant l'implémentation de Gensim : https://radimrehurek.com/gensim/models/phrases.html
9. Avec les paramètres : -size 300 -window 5 -negative 10 -hs 0 -sample 1e-5 -min-count 5

modèle	réf.	$R_{préc}$	MAP	P@1	P@2	P@5	P@10
fastText-wiki	para	9,9	6,0	36,5	29,9	21,3	15,9
	syn	15,5	18,4	21,9	15,7	9,2	5,8
BERT ctxt	para	9,5	5,7	36,5	30,4	22,4	17,0
	syn	15,6	17,9	21,8	16,0	9,5	6,1
BERT iso	para	7,4	4,4	30,9	26,2	19,6	14,6
	syn	14,0	15,8	19,2	14,6	8,7	5,5
BERT réfsyn	para	17,2	12,3	55,7	48,5	37,4	29,0
	syn	27,0	31,9	35,9	27,8	17,4	11,4

TABLE 1: Points de référence pour l'évaluation (valeurs x100)

Pour la mise en œuvre de la méthode d'injection, nous avons eu recours à la bibliothèque Sentence-Transformers [10]. Pour chaque relation (m_1, m_2), nous avons aussi considéré la relation (m_2, m_1), ce qui peut être vu comme une forme très simple d'augmentation de données. Comme le laisse apparaître l'équation 1, l'ensemble des relations à injecter est traité par lots, d'une taille de 512 relations chacun. L'apprentissage se fait en 10 époques, avec un taux d'apprentissage de $2e-5$, l'utilisation de l'optimiseur AdamW (Loshchilov & Hutter, 2019) et un nombre d'étapes d'échauffement égal à 10 % des relations à injecter, avec un schéma d'échauffement linéaire.

3.3 Évaluation des méthodes proposées

Points de référence. Le tableau 1 donne un certain nombre de références concernant notre évaluation du résultat de l'injection de relations sémantiques. La première de ces références, *fastText-wiki*, donne les performances en termes de similarité du modèle Skip-gram utilisé par Vulić *et al.* (2021) comme référence, appris à partir de la version anglaise de Wikipédia avec l'outil *fastText* (Bojanowski *et al.*, 2017). La deuxième, *BERT ctxt*, donne ces mêmes performances pour des plongements construits selon la méthode de Bommasani *et al.* (2020), c'est-à-dire en moyennant les représentations des occurrences des mots cibles apparaissant dans un ensemble de phrases. À l'instar de Ferret (2022), nous prenons 10 phrases par mot cible et les meilleurs résultats sont obtenus avec la couche L5. Comme on peut le constater, ces deux premières références sont très proches, indiquant au passage que concernant la similarité sémantique, modèles statiques et modèles contextuels sont très proches comme cela a déjà été observé par d'autres travaux (Lenci *et al.*, 2022). *BERT iso* correspond quant à elle au point départ de notre processus d'injection de relations (cf. section 2.1), les résultats du tableau 1 pour ce modèle étant issus de la couche L0. Le fait d'utiliser une seule occurrence sans contexte pour les mots cibles a clairement une incidence négative sur la performance comme le montre la comparaison avec *BERT ctxt* mais mobilise nettement moins de ressources. Notre dernière référence, *BERT réfsyn*, peut être considérée comme notre référence haute puisqu'elle correspond à l'injection faite dans un modèle BERT par Vulić *et al.* (2021) de 1 023 082 relations sémantiques issues de ressources constituées manuellement, en l'occurrence WordNet et le thésaurus Roget [11]. Les mesures sont données pour la couche 12 et le nombre d'époques dans ce cas est réduit à 2 compte tenu du nombre de relations.

10. https://www.sbert.net/
11. Il s'agit plus précisément de notre reproduction du travail de Vulić *et al.* (2021), dont le code n'est pas disponible.

	para	syn	# relations
BERT	13,9	5,2	17 007
CBERT	22,9	10,9	17 023
CBERT – têtes	24,2	11,8	13 465
CBERT – modifieurs	16,0	6,7	7 792
CBERT – ADJ NOM	26,2	12,4	10 511

TABLE 2: Exactitude (x100) des relations extraites par un modèle contextuel suivant le type de modèle et la structure des termes amorces

	P0	P1	P2	P3	P4	P5	P6	P7
para	32,0	30,3	24,9	31,1	30,9	31,7	26,5	31,7
syn	16,0	15,8	12,5	15,6	16,0	16,8	13,0	15,7

TABLE 3: Exactitude (x100) des relations extraites par les types formes d'amorces

Extraction de relations à partir d'un modèle contextuel. La méthode que nous avons proposée à la section 2.1 pour extraire des relations sémantiques à partir d'un modèle contextuel pose un certain nombre de questions auxquelles nous essayons de répondre ici en commençant par aborder, au travers des résultats du tableau 2, le problème du type de modèle et de la structure syntaxique des termes servant d'amorce. Les résultats sont donnés en termes d'exactitude des relations extraites par rapport à nos deux références. Il est à noter qu'ils ont été obtenus pour une forme générale d'amorce correspondant à P0 mais avec TERM et TERM_MSK faisant partie d'une même séquence. Le seuil d'information mutuelle minimale pour les termes extraits (cf. méthode de Mikolov *et al.* (2013b) évoquée ci-dessus) était par ailleurs égal à 10. Les deux premières lignes comparent le modèle BERT *base-uncased* avec le modèle CharacterBERT (El Boukkouri *et al.*, 2020), équivalent au modèle BERT base en termes de structure mais présentant la caractéristique de ne pas découper les mots en sous-mots. Cette comparaison permet de constater l'impact très notable de ce découpage des mots pour notre tâche d'extraction de relations, avec un très net avantage pour le modèle CharacterBERT, qui sera utilisé dans ce qui suit.

Les trois lignes suivantes ont trait quant à elles à la structure syntaxique des termes amorces et à la place qu'y occupe le mot cible. Nous constatons tout d'abord grâce aux deux première lignes que le mot cible en position de modifieur sur le plan syntaxique produit moins de relations qu'en position de tête et surtout, que ces relations sont beaucoup plus bruitées. Cette observation va dans le sens des travaux de Ferret (2013b), qui a montré qu'au sein de deux termes complexes entretenant un lien de similarité sémantique, il est plus probable d'avoir une relation de similarité sémantique entre les têtes syntaxiques des termes pour un même modifieur que le contraire, ce qui conduit à privilégier les termes ayant le mot cible pour tête syntaxique. La dernière ligne permet enfin de constater que la très grande majorité des relations sont obtenues à partir de la structure de terme ADJ NOM (NOM étant le mot cible), avec là encore des relations moins bruitées que pour les autres structures. Nous ne retiendrons donc pour TERM et TERM_MSK que des termes de type ADJ NOM.

Le tableau 3 permet pour sa part de juger des performances des différents types d'amorces présentés à la section 2.1, toujours avec un seuil d'information mutuelle minimale égal à 10 pour l'extraction des termes. Si la plupart de ces amorces donnent des résultats voisins, il faut remarquer que P2 et

relations	modèle	para	syn	# relations	# mots
extraites	statique	30,0	19,4	35 246	35 246
	contextuel	30,6	15,6	15 473	17 019
sélectionnées	statique	44,1	34,0	11 298	11 298
	contextuel	42,6	21,3	8 558	5 507
	fusion	41,1	24,2	18 430	14 199

TABLE 4: Exactitude (x100) et volumétrie des relations extraites puis sélectionnées

P6 obtiennent des résultats nettement inférieurs aux autres, sans qu'une raison très évidente puisse expliquer ce constat. La conclusion quant à la sensibilité par rapport à la forme des amorces est donc incertaine : si cette sensibilité n'est globalement pas très forte, elle peut être ponctuellement marquée, sans explication très claire. Dans ce qui suit, nous retiendrons l'amorce P0, la plus simple et une des deux meilleures.

Extraction et sélection de relations : synthèse. L'extraction des relations à partir d'un modèle statique telle que décrite à la section 2.1 ne demande de fixer que la taille du voisinage k et les mots cibles considérés. Dans le cas présent, nous avons retenu comme cibles les mots ayant une fréquence supérieure à 200 dans Wikipédia et une valeur $k = 1$ pour le voisinage. Il faut souligner à cet égard une différence importante entre les deux types de modèles : alors que le voisinage se limite au premier voisin pour le modèle statique, la qualité décroissant rapidement au-delà, nous l'étendons aux cinq premiers voisins pour le modèle contextuel, la dégradation de la qualité des relations étant beaucoup plus limitée à mesure de l'augmentation du rang.

Le tableau 4 évalue la qualité des relations extraites par nos deux types de modèles puis sélectionnées par réciprocité dans le graphe des *k-NN*, en regard avec leur volumétrie et le nombre de mots impliqués dans ces relations. Cette évaluation est faite sur la base de la présence de ces relations dans nos deux références, *syn* et *para*. Le modèle statique produit beaucoup plus de relations mais les deux types de modèles sont plus proches en termes de qualité, avec une équivalence sur l'ensemble des relations mais un avantage pour le modèle statique concernant les relations de synonymie. La sélection par réciprocité reproduit, et accentue même, ce biais initial pour ce qui est de la qualité des relations mais l'atténue fortement pour la volumétrie. Finalement, la fusion des deux ensembles de relations permet de constater leur complémentarité, avec un recoupement assez faible entre les deux et un niveau de performance plus proche du modèle contextuel que du modèle statique.

Enrichissement d'un modèle de langue. La dernière partie de notre évaluation concerne les résultats de l'injection des relations extraites dans un modèle de type BERT, illustrés par le tableau 5. Ce dernier rappelle notre point de départ, *BERT iso*, et donne les résultats pour chaque ensemble de relations extraites : celles issues du modèle statique, celles issues du modèle contextuel et la fusion de ces deux ensembles. Le premier constat est que l'injection des relations permet d'obtenir un gain de performance très significatif [12] par rapport à *BERT iso*, en particulier pour toutes les mesures P@r. Ce gain est globalement assez comparable pour les relations issues du modèle statique et celles issues du modèle contextuel. Il suit logiquement le même biais que les relations injectées. Il est ainsi plus important pour la synonymie dans le cas des relations du modèle statique et plus marqué pour

12. La significativité statistique des différences entre *BERT iso* et les autres modèles a été évaluée grâce à un test de Wilcoxon pour échantillons appariés avec $p < 0.01$.

modèle (couche)	réf.	$R_{préc}$	MAP	P@1	P@2	P@5	P@10
BERT iso (L0)	para	7,4	4,4	30,9	26,2	19,6	14,6
	syn	14,0	15,8	19,2	14,6	8,7	5,5
modèle statique (L11)	para	11,7	7,4	42,2	35,4	26,1	19,7
	syn	18,8	21,7	25,9	19,0	11,2	7,0
modèle contextuel (L11)	para	11,9	7,6	42,8	36,2	26,9	20,3
	syn	18,4	21,5	25,4	18,9	11,1	7,1
fusion (L12)	para	12,1	7,8	44,2	36,9	27,0	20,4
	syn	19,2	22,2	26,7	19,3	11,3	7,1

TABLE 5: Résultats de l'injection des relations sémantiques extraites (valeurs x100)

l'ensemble de nos relations sémantiques de référence en considérant les relations issues du modèle contextuel. La fusion des deux ensembles de relations permet d'obtenir un gain de performance supplémentaire pour nos deux références. Le niveau final atteint reste bien entendu en deçà du niveau constaté après l'injection de relations issues de ressources construites manuellement (cf. tableau 1) mais il est tout de même intéressant de constater qu'en l'absence de telles ressources, surtout dans les quantités utilisées par Vulić *et al.* (2021), il est tout de même possible d'enrichir sémantiquement un modèle de type BERT avec des relations sémantiques acquises automatiquement.

4 Conclusion et perspectives

Dans cet article, nous avons présenté à la fois deux méthodes pour extraire des relations sémantiques lexicales, l'une à partir d'un modèle de langue statique, l'autre à partir d'un modèle contextuel, et une méthode pour injecter ces relations afin d'enrichir sémantiquement un modèle contextuel. Les évaluations menées en termes de similarité sémantique ont montré le caractère effectif de la démarche proposée, qui permet de s'affranchir, au moins en partie, de relations sémantiques définies manuellement.

Outre le fait d'étendre les évaluations menées à d'autres tâches, un prolongement naturel de ce travail est d'étudier plus avant différentes formes d'amorces pour l'extraction de relations à partir de termes complexes et de comprendre plus précisément, notamment par l'observation des mécanismes d'attention, pourquoi certaines formes sont plus intéressantes que d'autres. Cette étude va par ailleurs de pair avec la prise en compte d'un ensemble plus vaste de structures de termes que la structure ADJ NOM principalement considérée ici. Le test d'autres modèles contextuels, tels que les modèles auto-régressifs, fait aussi partie des extensions très directes du travail présenté. Au-delà, nous souhaiterions également tester si un modèle contextuel enrichi sémantiquement comme nous l'avons réalisé pourrait conduire à une meilleure extraction de relations sémantiques, pouvant à son tour conduire à un meilleur enrichissement selon une procédure d'amorçage.

Remerciements

Nous remercions les relecteurs pour leur retour constructif. Ces travaux ont été réalisés grâce au supercalculateur Factory-IA financé par le Conseil Régional d'Île-de-France.

Références

BARONI M., DINU G. & KRUSZEWSKI G. (2014). Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, p. 238–247, Baltimore, Maryland.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146. DOI : 10.1162/tacl_a_00051.

BOMMASANI R., DAVIS K. & CARDIE C. (2020). Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, p. 4758–4781, Online.

BUDANITSKY A. & HIRST G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, **32**(1), 13–47.

CLARK K., KHANDELWAL U., LEVY O. & MANNING C. D. (2019). What does BERT look at ? an analysis of BERT's attention. In *2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 276–286, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-4828.

CLAVEAU V., KIJAK E. & FERRET O. (2014). Improving distributional thesauri by exploring the graph of neighbors. In *25th International Conference on Computational Linguistics (COLING 2014)*, p. 709–720, Dublin, Ireland.

CURRAN J. R. & MOENS M. (2002). Improvements in automatic thesaurus extraction. In *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, p. 59–66, Philadelphia, USA.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2019)*, p. 4171–4186, Minneapolis, Minnesota. DOI : 10.18653/v1/N19-1423.

EL BOUKKOURI H., FERRET O., LAVERGNE T., NOJI H., ZWEIGENBAUM P. & TSUJII J. (2020). CharacterBERT : Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *28th International Conference on Computational Linguistics (COLING 2020)*, p. 6903–6915, Barcelona, Spain (Online : International Committee on Computational Linguistics.

ETHAYARAJH K. (2019). How Contextual are Contextualized Word Representations ? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, p. 55–65, Hong Kong, China. DOI : 10.18653/v1/D19-1006.

FARUQUI M., DODGE J., JAUHAR S. K., DYER C., HOVY E. & SMITH N. A. (2015). Retrofitting Word Vectors to Semantic Lexicons. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT 2015)*, p. 1606–1615, Denver, Colorado.

FARUQUI M., TSVETKOV Y., RASTOGI P. & DYER C. (2016). Problems with evaluation of word embeddings using word similarity tasks. In *Workshop on Evaluating Vector-Space Representations for NLP (RepEval 2016)*, p. 30–35, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/W16-2506.

FERRET O. (2013a). Identifying bad semantic neighbors for improving distributional thesauri. In *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, p. 561–571.

FERRET O. (2013b). Sélection non supervisée de relations sémantiques pour améliorer un thésaurus distributionnel. In *20ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2013)*, p. 48–61, Les Sables d'Olonne, France.

FERRET O. (2022). Building static embeddings from contextual ones : Is it useful for building distributional thesauri ? In *13th Language Resources and Evaluation Conference (LREC 2022)*, p. 2583–2590, Marseille, France.

FOX E. A. & SHAW J. A. (1994). Combination of multiple searches. In *2nd Text REtrieval Conference (TREC-2)*, volume 243 : NIST.

GREFENSTETTE G. (1994). *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers.

HARRIS Z. S. (1954). Distributional Structure. *Word*, **10**(2-3), 146–162.

HENDERSON M., VULIĆ I., GERZ D., CASANUEVA I., BUDZIANOWSKI P., COOPE S., SPITHOURAKIS G., WEN T.-H., MRKŠIĆ N. & SU P.-H. (2019). Training neural response selection for task-oriented dialogue systems. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, p. 5392–5404, Florence, Italy. DOI : 10.18653/v1/P19-1536.

JOHNSON J., DOUZE M. & JÉGOU H. (2021). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, **7**(3), 535–547.

LAUSCHER A., VULIĆ I., PONTI E. M., KORHONEN A. & GLAVAŠ G. (2020). Specializing unsupervised pretraining models for word-level semantic similarity. In *28th International Conference on Computational Linguistics (COLING 2020)*, p. 1371–1383, Barcelona, Spain (Online). DOI : 10.18653/v1/2020.coling-main.118.

LEE J. H. (1997). Analyses of multiple evidence combination. In *20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, p. 267—-276, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/258525.258587.

LENCI A., SAHLGREN M., JEUNIAUX P., CUBA GYLLENSTEN A. & MILIANI M. (2022). A comparative evaluation and analysis of three generations of Distributional Semantic Models. *Language Resources and Evaluation*. DOI : 10.1007/s10579-021-09575-z.

LIN D. (1998). Automatic retrieval and clustering of similar words. In *17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (ACL-COLING'98)*, p. 768–774, Montréal, Canada.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv :1907.11692*.

LOSHCHILOV I. & HUTTER F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

MANNING C. D., SURDEANU M., BAUER J., FINKEL J., BETHARD S. J. & MCCLOSKY D. (2014). The Stanford CoreNLP natural language processing toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics : System Demonstrations (ACL 2014)*, p. 55–60.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, p. 3111–3119.

MILLER G. A. (1990). WordNet : An On-Line Lexical Database. *International Journal of Lexicography*, **3**(4).

NASEEM U., RAZZAK I., KHAN S. K. & PRASAD M. (2021). A Comprehensive Survey on Word Representation Models : From Classical to State-of-the-Art Word Representation Language Models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, **20**(5), 74 :1–74 :35. DOI : 10.1145/3434237.

PADRÓ M., IDIART M., VILLAVICENCIO A. & RAMISCH C. (2014a). Comparing similarity measures for distributional thesauri. In *Ninth International Conference on Language Resources and Evaluation (LREC'14)*, p. 2964–2971, Reykjavik, Iceland : European Language Resources Association (ELRA).

PADRÓ M., IDIART M., VILLAVICENCIO A. & RAMISCH C. (2014b). Nothing like good old frequency : Studying context filters for distributional thesauri. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, p. 419–424, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1047.

QIANG J., LI Y., ZHU Y., YUAN Y. & WU X. (2020). *Lexical Simplification with Pretrained Encoders*. Rapport interne, arXiv preprint arXiv :1907.06226.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). *Improving language understanding by generative pre-training*. Rapport interne, OpenAI.

REIMERS N. & GUREVYCH I. (2019). Sentence-BERT : Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3982–3992, Hong Kong, China. DOI : 10.18653/v1/D19-1410.

SHAH S., REDDY S. & BHATTACHARYYA P. (2020). A retrofitting model for incorporating semantic relations into word embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 1292–1298, Barcelona, Spain (Online). DOI : 10.18653/v1/2020.coling-main.111.

VULIĆ I., PONTI E. M., KORHONEN A. & GLAVAŠ G. (2021). LexFit : Lexical fine-tuning of pretrained language models. In *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJNLP 2021)*, p. 5269–5283, Online. DOI : 10.18653/v1/2021.acl-long.410.

VULIĆ I., PONTI E. M., LITSCHKO R., GLAVAŠ G. & KORHONEN A. (2020). Probing Pretrained Language Models for Lexical Semantics. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, p. 7222–7240, Online. DOI : 10.18653/v1/2020.emnlp-main.586.

WANG Y. (2022). *Exploration des relations terminologiques entre les termes multi-mots dans les modèles de sémantique distributionnelle*. Thèse de doctorat, Université Toulouse-Jean Jaurès.

WANG Y., WANG W., CHEN Q., HUANG K., NGUYEN A., DE S. & HUSSAIN A. (2023). Fusing external knowledge resources for natural language understanding techniques : A survey. *Information Fusion*, **92**, 190–204. DOI : https ://doi.org/10.1016/j.inffus.2022.11.025.

WU S., CRESTANI F. & BI Y. (2006). Evaluating score normalization methods in data fusion. In *Third Asia Conference on Information Retrieval Technology (AIRS'06)*, p. 642–648 : Springer-Verlag.

ZHOU W., GE T., XU K., WEI F. & ZHOU M. (2019). BERT-based lexical substitution. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, p. 3368–3373, Florence, Italy. DOI : 10.18653/v1/P19-1328.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Géométrie de l'auto-attention en classification : quand la géométrie remplace l'attention

Loïc Fosse Duc Hau Nguyen Pascale Sébillot Guillaume Gravier
Univ Rennes, CNRS, Inria, INSA Rennes – IRISA, Campus de Beaulieu, 35042 Rennes
loic.fosse@insa-rennes.fr, duc-hau.nguyen@irisa.fr,
pascale.sebillot@irisa.fr, guig@irisa.fr

RÉSUMÉ

Plusieurs études ont mis en évidence l'anisotropie des plongements issus d'un modèle BERT au sein d'un énoncé, c'est-à-dire leur concentration dans une direction donnée, notamment dans une tâche de classification. Dans cet article, nous cherchons à mieux comprendre ce phénomène et comment cette convergence se construit en analysant finement les propriétés géométriques des plongements, des clés et des valeurs dans une couche d'auto-attention. Nous montrons que la direction vers laquelle les plongements s'alignent caractérise la classe d'appartenance de l'énoncé. Nous étudions ensuite le fonctionnement intrinsèque de la couche d'auto-attention et les mécanismes en jeu entre clés et valeurs pour garantir la construction d'une représentation anisotrope. Cette construction se fait de manière progressive lorsque plusieurs couches sont empilées. Elle s'avère également robuste à des contraintes externes sur la distribution des poids d'attention, compensées par le modèle en jouant sur les valeurs et les clés.

ABSTRACT

Geometry of self-attention in classification: when geometry replaces attention

Various studies have highlighted the anisotropy of BERT word embeddings within an utterance, i.e., their concentration in a given direction, especially in a classification task. In this paper, we aim at better understanding this phenomenon and how this convergence is built by analyzing the geometric properties of the word embeddings, keys and values within a self-attention layer. We show that the direction towards which embeddings align themselves characterizes class membership. We then study the intrinsic mechanism of the self-attention layer and the mechanisms at play between keys and values to ensure the construction of an anisotropic representation. This construction is progressive when several layers are stacked. It also proves to be robust to external constraints on the distribution of attention weights, which the model compensates through the values and keys.

MOTS-CLÉS : classification, auto-attention, transformers, bertologie.

KEYWORDS: text classification, self-attention, transformers, bertology.

1 Introduction

Les modèles transformeurs fondés sur le mécanisme d'auto-attention sont au cœur de l'état de l'art en traitement automatique des langues dans de nombreuses tâches. Ce succès est en partie dû à l'existence de modèles pré-entraînés de manière générique comme modèle de langue (causal ou masqué) qui sont ensuite adaptés à une tâche précise (Radford *et al.*, 2018; Devlin *et al.*, 2019;

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 137–150.

Conneau & Lample, 2019; Brown *et al.*, 2020).

Afin de mieux appréhender le fonctionnement interne de ces modèles, plusieurs études se sont intéressées aux différentes informations linguistiques portées par les modèles pré-entraînés, typiquement en utilisant les plongements aux différents niveaux d'un transformeur pré-entraîné, pour mesurer leur performance dans différentes tâches linguistiques. On peut par exemple citer (Van Aken *et al.*, 2019; Htut *et al.*, 2019; Jawahar *et al.*, 2019; Kovaleva *et al.*, 2019; Lin *et al.*, 2019) dont les principaux résultats sont résumés par Rogers *et al.* (2020). Ces travaux montrent que les plongements aux différents niveaux encodent des informations linguistiques différentes, de plus en plus riches et complexes au fur et à mesure que l'on monte en abstraction, les dernières couches étant quant à elle très influencées par la tâche utilisée pour le pré-entraînement. Certaines études se sont également penchées sur les propriétés géométriques des plongements aux différents niveaux d'un modèle transformeur (Reif *et al.*, 2019; Ethayarajh, 2019; Hernandez & Andreas, 2021; Fosse *et al.*, 2022). Plusieurs d'entre elles mettent clairement en avant l'anisotropie croissante des plongements des *tokens* d'une même phrase, c'est-à-dire leur concentration dans une direction au travers des différentes couches d'attention (Ethayarajh, 2019; Cai *et al.*, 2021; Fosse *et al.*, 2022). En particulier, Fosse *et al.* (2022) mettent en évidence une concentration forte des plongements dans une même direction lorsqu'un modèle pré-entraîné est adapté pour la classification de texte, tâche pour laquelle la distinction entre les *tokens* n'a au final plus d'importance.

Même si Cai *et al.* (2021) montrent qu'on peut retrouver une isotropie permettant de distinguer les différents *tokens* dans des sous-espaces, cette dernière observation soulève des questions sur la manière dont les couches d'auto-attention construisent cette convergence dans une direction unique, questions au centre de cet article.

De manière intéressante, ces observations sur la géométrie des plongements sont à mettre en parallèle avec les travaux menés sur les poids d'attention (Clark *et al.*, 2019; Bai *et al.*, 2021; Bibal *et al.*, 2022) qui soulèvent de nombreux débats sur l'intérêt de ces poids pour expliquer la décision du modèle (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019; Bibal *et al.*, 2022). De nombreux exemples visuels montrent qu'il est parfois possible d'interpréter *a posteriori* les poids d'attention [1]. Cependant, plusieurs observations troublantes viennent mettre à mal la possibilité d'utiliser ces poids comme explication. D'une part, il a été mis en évidence que la distribution des poids d'attention sur les *tokens* d'une phrase a tendance à être répartie de manière uniforme dans un modèle d'attention associé à un LSTM (Mohankumar *et al.*, 2020; Nguyen *et al.*, 2021), ou, dans des modèles transformeurs, à se concentrer sur des *tokens* peu informatifs comme [CLS] (Clark *et al.*, 2019). Il a également été mis en évidence qu'il est possible de modifier les poids d'attention sans réelle incidence sur la classification (Voita *et al.*, 2019; Jain & Wallace, 2019; Pruthi *et al.*, 2020). Cette propriété est utilisée dans plusieurs travaux pour contraindre les poids d'attention. On peut, par exemple, chercher à les rendre plus parcimonieux, et donc plus plausibles dans l'explication de la décision (Niculae & Blondel, 2017; Mohankumar *et al.*, 2020; Nguyen *et al.*, 2022). Une autre orientation consiste à contraindre les poids d'attention pour guider l'apprentissage ou la génération d'une explication (Nguyen & Nguyen, 2018; McGuire & Tomuro, 2021; Carton *et al.*, 2022; Paranjape *et al.*, 2020).

Ces deux constatations – la convergence intra-phrase des plongements et la tendance à une distribution uniforme des poids d'attention – nous amène dans cet article à nous interroger plus avant sur le rapport entre la géométrie des plongements et la tâche de classification. Au travers d'une analyse théorique et expérimentale du mécanisme d'auto-attention, nous cherchons à déterminer si la convergence des plongements est une propriété intrinsèque des transformeurs et à comprendre comment cette

1. Voir par exemple la figure 1 dans (Clark *et al.*, 2019).

convergence se construit en jouant sur les clés et les valeurs du modèle.

Dans ce travail, nous mettons en évidence que, dans les tâches de classification d'énoncés, l'anisotropie des plongements au sein d'une phrase est liée à la classe : en d'autres termes, les plongements s'alignent dans une direction qui dépend de la classe, ce qui correspond au fonctionnement intrinsèque du modèle. Nous mettons ensuite en évidence le jeu entre clé et valeur qui permet de construire cet alignement des plongements au travers des couches d'auto-attention, y compris lorsque des contraintes externes sont imposées sur la distribution des poids d'attention. Dans ce dernier cas, nous montrons comment le modèle compense les contraintes pour assurer son fonctionnement en alignant dans une direction les plongements.

2 Formalisme, tâches

Nous posons tout d'abord le formalisme du mécanisme d'auto-attention, au cœur des modèles récents pré-entraînés comme modèle de langage. Nous décrivons ensuite succinctement les modèles et les jeux de données utilisés pour les expériences.

2.1 Formalisme et géométrie de l'auto-attention

Le principe fondamental de ces modèles consiste à transformer une séquence de plongements $\mathbf{x} = \{x_1, \ldots, x_n\}$, $x_i \in \mathbb{R}^d$, en une nouvelle séquence $\mathbf{y} = \{y_1, \ldots, y_n\}$, $y_i \in \mathbb{R}^d$, la nouvelle représentation y_i étant obtenue à partir de l'attention a_{ij} qu'il faut porter à chacun des *tokens* ($j \in [1, n]$) de la séquence en entrée pour obtenir la nouvelle représentation du *token* i.

En pratique, chaque *token* x_i est projeté dans 3 espaces distincts, l'espace des clés, celui des valeurs et celui de requêtes. On notera $q_i = x_i\ \mathbf{Q}$ la clé à la position i et $k_i = x_i\ \mathbf{K}$ (resp. $v_i = x_i\ \mathbf{V}$) la clé (resp. la valeur). Les trois matrices $\mathbf{Q}$, $\mathbf{K}$ et $\mathbf{V}$, chacune de dimension $d \times d$, jouent le rôle de projection du plongement d'un *token* vers un nouvel espace de même dimension que l'espace d'entrée et constituent les principaux paramètres du modèle. Pour la position i, l'attention à porter à chacun des *tokens* de la phrase pour modifier x_i est donnée par

$$a_{ij} = \frac{\exp\left(q_i \cdot k_j/\sqrt{d}\right)}{\sum_{l=1}^{n} \exp\left(q_i \cdot k_l/\sqrt{d}\right)} \ . \tag{1}$$

L'ensemble des $a_{ij} \in \mathbb{R}^+$ constitue les poids d'attention et permet de définir la nouvelle représentation du *token* i selon

$$y_i = \sum_{j=1}^{n} a_{ij} v_j \ . \tag{2}$$

Il est intéressant de donner à ce stade une interprétation géométrique de ce mécanisme par rapport aux observations de l'introduction. Chaque y_i étant obtenu par une combinaison convexe des v_j à partir de poids d'attention positifs, si les v_j se concentrent dans une direction, y_i ne peut par définition se retrouver que dans le cône défini par les v_j comme illustré à la figure 1. De manière duale, on note que si l'ensemble des clés k_j se concentrent dans un cône, alors les poids d'attention tendent vers

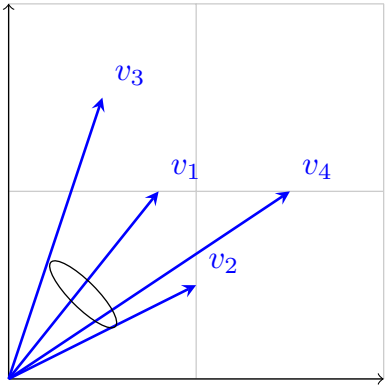


FIGURE 1 – Interprétation géométrique de l'équation 2 : illustration d'un cône défini par l'ensemble des valeurs v_j, $j \in [1, n]$.

une distribution uniforme quelque soit la requête q_i. Enfin, si les requêtes s'alignent, alors les poids d'attention a_{ij} sont identiques pour tous les *tokens* x_i.

2.2 Modèles fondés sur l'auto-attention

Ce mécanisme d'auto-attention correspond à une tête d'attention dans un modèle transformeur (Vaswani *et al.*, 2017), en particulier dans la partie encodeur qui nous intéresse. En pratique, ces modèles s'appuient sur plusieurs couches *transformer* qui modifient progressivement les représentations des *tokens* en entrée du modèle. Chaque couche *transformer* se compose de deux opérations successives : un ensemble de têtes d'attention qui opèrent en parallèle sur la représentation en entrée et dont les sorties sont combinées par concaténation et projection ; une projection point à point de chacun des n vecteurs issus de l'étape précédente. Une connexion résiduelle est utilisée autour de chacune de ces opérations qui sont également chacune suivie d'une normalisation des vecteurs (*layer normalization*). En pratique, chaque tête d'attention opère sur un sous-ensemble des dimensions de l'espace dans lequel les *tokens* sont représentés, la combinaison se limitant alors à une concaténation des représentations issues de chaque tête. Malgré les différentes opérations et l'utilisation de plusieurs têtes, le mécanisme d'attention avec les propriétés géométriques que nous avons défini précédemment reste l'opération centrale au sein d'une couche *transformer*.

La première partie de notre étude s'appuie sur le modèle BERT standard (Devlin *et al.*, 2019) [2], qui comporte 12 couches, chacune composée de 12 têtes d'attention. Chacune des têtes est en charge d'un sous-espace de l'espace complet, soit 12 têtes de dimension 64 pour une dimension totale de 768. Les paramètres du modèle ont été estimés sur un grand volume de données par le biais d'une tâche de modèle de langage avec masque. Lorsque ce modèle est utilisé pour une tâche de classification, une couche de classification est ajoutée et prend en entrée la représentation contextualisée par le modèle BERT du *token* [CLS] de début d'énoncé afin de prédire la classe de l'énoncé. L'ensemble des paramètres du modèle (couche de classification et l'ensemble des couches d'attention) est réestimé par quelques itérations selon un critère standard de minimisation de l'entropie croisée.

Dans la seconde partie de notre étude où nous nous intéressons au fonctionnement intrinsèque des têtes d'attention, nous utilisons une réimplémentation des têtes d'attention dont les paramètres sont estimés

2. Nous utilisons le *checkpoint* `bert-base-uncased` dans le dépôt Huggingface.

modèle	YelpHat	HateXplain	e-SNLI
BERT adapté	93,6	40,1	88,7
auto-attention	91,3	61,1	63,1

TABLE 1 – Performance de classification (en *% d'accuracy*) pour le modèle BERT adapté et pour un modèle avec une couche d'auto-attention.

directement pour la tâche de classification. Le bloc d'attention inclut une couche d'auto-attention à une seule tête (équations 1 et 2) avec une connexion résiduelle et suivi d'une normalisation. Le modèle final de classification utilisé est similaire à celui du paragraphe précédent.

2.3 Tâches et données

Dans cet article, nous nous intéressons au fonctionnement du mécanisme d'attention dans les tâches de classification de texte. Nous exploitons plusieurs jeux de données standards dans le domaine de la classification, en particulier les données liées au *benchmark* ERASER (DeYoung *et al.*, 2020) qui permettent ensuite d'étudier les poids d'attention à l'aune d'une annotation humaine des *tokens* importants pour la classification [3]. Nous utilisons en particulier trois *corpora* en anglais, du plus facile au plus difficile :

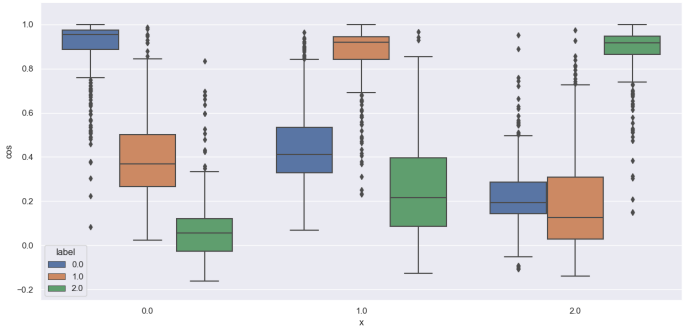
— YelpHat (Sen *et al.*, 2020) vise à la classification en polarité d'avis concernant des restaurants. Le corpus est composé d'environ 3.5k avis pour l'apprentissage et une quantité équivalente pour le test.

— HateXplain (Mathew *et al.*, 2021) a été conçu pour une tâche de classification de messages postés sur des réseaux sociaux en trois catégories : haineux, agressif, neutre. Le corpus est composé de 15k messages pour l'apprentissage et environ 2k pour pour la validation et le test respectivement.

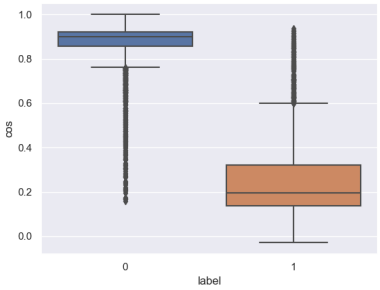
— e-SNLI (Camburu *et al.*, 2018), une extension du corpus SNLI (Bowman *et al.*, 2015), permet une tâche d'inférence linguistique qui consiste à déterminer si deux énoncés constituent une suite logique, sont en opposition ou n'ont aucun rapport. Le corpus contient environ 550k paires de phrase pour l'apprentissage, et environ 10k paires pour la validation et pour le test respectivement. Pour la tâche de classification, les deux énoncés sont concaténés en ajoutant un *token* [SEP] entre eux.

Pour tous ces corpus, la classification est réalisée à partir du plongement contextualisé du *token* [CLS] en utilisant une couche dense de projection. Lors de l'adaptation à la tâche de modèles pré-entrainés, l'ensemble des paramètres sont réestimés sur quelques itérations. L'adaptation à la tâche est effectuée avec les paramètres suivants : graine aléatoire ; optimiseur Adam sur 50 itérations (*epochs*) avec un taux d'apprentissage de 5×10^{-5} pour BERT et de 0.001 pour le second modèle ; stratégie d'arrêt précoce au bout de 5 itérations sans amélioration. Nous rapportons dans le tableau 1 les performances sur les trois corpus, avec le modèle BERT adapté ainsi qu'avec un seul bloc d'attention. On notera simplement les points suivants : le modèle BERT adapté pour HateXplain donne de mauvais résultats, ce qui est cohérent avec plusieurs résultats rapportés sur Internet ; le corpus YelpHat étant facile, l'écart entre BERT pré-entraîné et un modèle avec un seul bloc d'attention est limité.

3. Cette dernière analyse n'entre pas dans le cadre de cette étude mais les données ont été choisies pour pouvoir attaquer cette problématique par la suite.



(a) e-SNLI



(b) YelpHat

FIGURE 2 – Distribution de la similarité cosinus entre le plongement d'un énoncé représentant une classe (r_i) et les plongements du reste des énoncés du corpus (cls_j) en fonction de leur classe pour e-SNLI (2a) et YelpHat (2b).

3 Une convergence en lien avec la classification

Nous étudions tout d'abord la géométrie des plongements issus du modèle BERT adapté à la tâche. Comme dans les études précédentes (Reif *et al.*, 2019; Ethayarajh, 2019; Hernandez & Andreas, 2021; Fosse *et al.*, 2022), nous observons bien évidemment le renforcement de l'anisotropie des plongements au sein d'une même phrase lorsque l'on traverse les couches du modèle, avec une concentration des plongements au sein d'un (hyper)cône dans l'espace des plongements. En effet, si l'on mesure la similarité cosinus moyenne entre les plongements de deux *tokens* au sein d'une même phrase sur la tâche e-SNLI, elle croît de manière *quasi* systématique dans le modèle pré-entraîné, à l'exception notable de la dernière couche. La similarité moyenne passe ainsi progressivement de 0.12 en entrée du modèle à 0.41 en sortie, un maximum (proche de 0.42) étant atteint à la sortie de l'avant-dernière couche. Sur le modèle après adaptation à la tâche, la convergence est nettement plus marquée, passant de 0.08 en entrée à 0.69 en sortie, avec une croissance constante et particulièrement marquée sur les 4 dernières couches où la similarité moyenne passe de 0.34 à 0.69. Un comportement similaire est observé sur d'autres corpus, notamment YelpHat et IMDb (Maas *et al.*, 2011), corroborant ainsi les observations de (Fosse *et al.*, 2022).

Ces observations sont faites énoncé par énoncé et sont valables pour les plongements au sein d'une même phrase. Dans le prolongement, nous nous intéressons donc à l'analyse de ce phénomène d'anisotropie au travers des phrases. Une hypothèse assez naturelle que nous souhaitons vérifier est que, dans le cas d'un modèle BERT adapté à une tâche de classification, la direction prise par les plongements au sein d'un énoncé est caractéristique de la classe de l'énoncé.

Nous vérifions cette hypothèse au travers de l'expérience suivante. Pour chaque classe i dans un corpus, nous choisissons un représentant de cette classe, r_i, représenté par le plongement final du *token* [CLS]. Ce représentant est pris au hasard parmi les énoncés de la classe i qui sont bien classés par le modèle. Nous noterons par la suite cls_j le plongement de [CLS] correspondant au j-ème énoncé du corpus : nous avons donc $r_i = \text{cls}_{i^*}$, i^* étant l'indice d'énoncé du représentant choisi pour la classe i. Pour l'ensemble des énoncés $j \neq i^*$ du corpus, nous mesurons la similarité cosinus entre r_i et cls_j de manière à analyser la distribution des similarités $\cos(r_i, \text{cls}_j)$ en fonction de la classe de l'énoncé j.

phrase	+/-	cos
Last summer I had an appointment to get new tires and had to wait a super long time. I also went in this week for them to fix a minor problem with a tire they put on. They "fixed" it for free, and the very next morning I had the same issue. I called to complain, and the "manager" didn't even apologize ! ! ! So frustrated. Never going back. They seem overpriced, too.	-	
Been coming to cafe rio for awesome fast Mexican food for years. Lived in Utah for a while...just like you Camilla k. I loved the ones in Utah and thought I would give this one a chance. The manager guy, don't know his name, is a total jerk. The food wasn't even warm. It was cold. My wife and I are sitting here right now and I'm so upset that I have to leave this review right now. Just awful service and not even good food anymore. Gradually getting worse. I'm going to costa vida from now on.	-	0,96
SO GOOD ! ! ! ! ! ! ! The only roll I got that wasn't good was a lobster roll. It just had no flavor. Everything else I had was AMAZING ! ! ! Now that I'm done raving about the food, I do have two complaints. 1) The hostess wasn't super friendly or anything. She was really hard to understand and made no effort to speak more clearly so we'd know what she was talking about. 2) There's no where near enough tables. The place is an okay size but there's probably only like 10 tables ? Maybe I'm remembering wrong. There's also no sushi bar, which I don't care about, but some people do.	+	0,93
I am a huge steak person. I live in LA and I've been to every fancy steakhouse in LA, most of the well known steakhouses in NY and Vegas. and Rig may not be the best steakhouse I've ever been to, but it's up there. What truly impressed me was the value. The steaks and appetizers were as good as Mastro's (both Vegas and LA) yet it cost almost half as much. The service was as good. 've told my wife that we'll be going there every time we go to Vegas.	+	0,2
Expensive Gringo Mexican food. Saving grace is the setting. Wonderful pond with ducks in the middle of the facility. Go for the beauty of it, not for the "Mexican" food. They seem to cater to the Cave Creek tourist "semi cowboy" trade.	-	0,09

TABLE 2 – Exemples d'énoncés positifs (+) et négatifs (-) du corpus YelpHat et de leur distance à un énoncé négatif représentatif (ligne 1).

La figure 2 montre sur la partie gauche les distributions obtenues pour chacune des trois classes de e-SNLI (resp. suite logique, sans rapport, contradiction) et, sur la partie droite, celles obtenues pour la classe négative (0) du corpus YelpHat. Ainsi, pour la figure 2b, on observe nettement que les plongements cls_j de la classe négative présentent très majoritairement une grande similarité avec l'énoncé représentatif de cette classe, tandis que les plongements cls_j pour la classe positive sont presque majoritairement orthogonaux. De manière similaire, on voit sur la figure 2a que la direction des plongements cls_j correspond bien aux classes. On note au passage sur cette dernière figure que si le trois classes sont relativement séparables par rapport à un représentant des classes suite logique ou sans rapport, ces deux dernières classes sont difficilement séparables du point de vue d'un représentant de la classe contradiction.

Ces résultats mettent donc clairement en évidence que la direction vers laquelle le modèle plonge la représentation des *tokens* en entrée est l'élément qui caractérise la classe du document. Ceci traduit le fonctionnement inhérent du modèle pour réaliser sa tâche et constitue donc un comportement souhaitable. Nous avons également vérifié que cette propriété est bien induite par l'adaptation du modèle pré-entraîné : la même analyse des distributions des similarités cosinus effectuée sur le modèle pré-entraîné montre que, dans ce cas, il n'est pas possible de distinguer les classes.

Enfin, nous complétons ces observations par quelques exemples d'énoncés positifs (+) et négatifs (-) issus du corpus YelpHat dans le tableau 2, accompagnés de leur distance (cos) à un énoncé

représentatif de la classe négative (ligne 1) : le premier groupe correspond aux énoncés (resp. - et +) les plus proches de l'énoncé de la ligne 1 ; le second groupe donne l'exemple médian pour la classe + [4] et l'exemple le plus éloigné pour la classe -. On note que la représentation issue du modèle BERT dépasse largement le niveau lexical, avec peu de mots porteurs de polarité en commun dans les énoncés proches du représentant. L'exemple positif le plus proche s'explique en particulier par les aspects négatifs que l'énoncé contient en dépit d'un avis globalement positif.

Ces observations sont à mettre en relation avec deux éléments discutés dans l'introduction. D'une part, elle apporte des éléments au débat sur l'attention comme vecteur d'explicabilité, du moins dans le cas de la classification de texte : si la convergence forte des plongements au sein d'une phrase est bénéfique, voire indispensable à la classification, elle laisse peu d'espoir à l'obtention de quelques *tokens* explicatifs grâce à l'attention. En effet, du fait de la convergence, les requêtes et clés dérivées des plongements convergent également dans une même direction, résultant ainsi en une attention *quasi* uniformément distribuée sur les *tokens*, comme observé dans plusieurs travaux (Mohankumar *et al.*, 2020; Nguyen *et al.*, 2021). D'autre part, les travaux visant à contraindre les poids d'attention pour les rendre parcimonieux – et donc potentiellement plus adaptés à une explication – montrent un succès limité (Niculae & Blondel, 2017; Nguyen & Nguyen, 2018; Mohankumar *et al.*, 2020; McGuire & Tomuro, 2021; Nguyen *et al.*, 2022; Sasaki *et al.*, 2023). La convergence des plongements dans une direction dépendante de la classe éclaire ce débat d'une explication possible, à laquelle nous nous intéressons par la suite : si cette notion de projection relève du fonctionnement intrinsèque des modèles, alors ce dernier compense les contraintes imposées sur l'attention pour permettre la convergence des plongements dans la direction la plus appropriée, et cela d'autant plus facilement que le modèle possède un grand nombre de paramètres.

4 Un jeu entre clés, valeurs et attention

Au vu des résultats précédents, nous nous intéressons à voir si l'alignement des plongements dans une direction représentative de chaque classe dans une tâche de classification est inhérente à la couche d'auto-attention et, le cas échéant, comment l'auto-attention permet de construire cette convergence. Nous nous affranchissons pour cela du modèle pré-entraîné comme modèle de langue et utilisons ici un modèle simplifié à une seule tête d'auto-attention et dont les paramètres peuvent être estimés directement pour la tâche de classification.

4.1 Convergence des clés et des valeurs : un mécanisme inhérent au fonctionnement de la couche d'auto-attention

Nous nous sommes tout d'abord intéressés à comprendre le rôle joué par les clés, les valeurs et, dans une moindre mesure, les requêtes pour construire la convergence des plongements au travers des couches. Nous mesurons en particulier la convergence des clés et des valeurs au sein d'un même énoncé. Pour cela, nous introduisons une métrique de similarité des éléments d'une matrice, largement inspirée de la mesure de conicité de (Ethayarajh, 2019), et donnée par

$$\text{sim}(A) = \frac{2}{n(n-1)} \sum_{i<j} \left[(AA^t) . \frac{1}{\sqrt{\text{diag}(AA^t)\text{diag}(AA^t)^t}} \right] (i,j) \tag{3}$$

4. similarité correspondant à la médiane de la distribution des similarités des exemples positifs dans la figure 2b

		HatexPlain					YelpHat					E-SNLI	
		l=1	**l=2**	**l=3**	**l=4**	**l=5**	**l=1**	**l=2**	**l=3**	**l=4**	**l=5**	**l=1**	**l=2**
	L=1	0.713	-	-	-	-	0.578	-	-	-	-	0.657	-
	L=2	0.691	0.683	-	-	-	0.649	0.556	-	-	-	0.719	0.489
A=K	L=3	0.698	0.688	0.841	-	-	0.597	0.431	0.537	-	-	-	-
	L=4	0.614	0.620	0.748	0.840	-	0.714	0.717	0.702	0.860	-	-	-
	L=5	0.624	0.647	0.777	0.913	0.973	0.584	0.542	0.761	0.816	0.959	-	-
	L=1	0.542	-	-	-	-	0.372	-	-	-	-	0.524	-
	L=2	0.510	0.740	-	-	-	0.409	0.511	-	-	-	0.182	0.746
A=V	L=3	0.605	0.688	0.88	-	-	0.461	0.371	0.494	-	-	-	-
	L=4	0.592	0.561	0.785	0.931	-	0.429	0.613	0.803	0.904	-	-	-
	L=5	0.606	0.673	0.858	0.958	0.992	0.417	0.624	0.780	0.9023	0.972	-	-

TABLE 3 – Mesure de la moyenne des similarité intra-énoncé des clés (K) et des valeurs (V). Les lignes (L) représentent le nombre de couches du modèle, les colonnes (l) représentent la couche à laquelle les mesures sont effectuées.

où $A \in \mathbb{R}^{n \times d}$ est une matrice regroupant un ensemble de n clés ou de n valeurs de dimension d, $\text{diag}(A)$ désignant la diagonale de la matrice A. Une valeur de 1 correspond à une matrice A dont les lignes sont toutes co-linéaires, 0 correspondant à des vecteurs lignes orthogonaux, et -1 à des vecteurs lignes alignés dans des directions opposés [5]. En d'autres termes, $\text{sim}(A)$ mesure à quelle point les lignes de la matrice se concentrent dans une direction, mesurant ainsi l'étroitesse du cône défini par les clés ou les valeurs au sein d'une phrase.

Pour chacun des trois corpus e-SNLI, YelpHat et HateXplain, nous avons entraîné des modèles simplifiés à une tête d'auto-attention (plus connexion résiduelle et normalisation – *cf*. section 2.2) avec un nombre de couches variant de 1 à 5 pour YelpHat et HateXplain et, pour des raisons de temps de calcul, de 1 à 2 pour e-SNLI. Nous mesurons pour chacun de ces modèles et au niveau de chacune des couches la valeur moyenne de la similarité des clés et des valeurs donnée par l'équation 3 au sein d'une phrase. La moyenne est effectuée sur 1 000 énoncés choisis aléatoirement dans les données de test en respectant l'équilibre des classes. L'ensemble de ces résultats est donné dans le tableau 3 où les lignes (L) indiquent le nombre de couches du modèle, chaque colonne (l) correspondant à la mesure de similarité prise au niveau de la couche l. Le premier ensemble de valeurs correspond à la similarité moyenne des clés au sein d'une phrase ($A = K$), le second à celle des valeurs ($A = V$).

Les résultats montrent clairement que la convergence se construit bien au niveau de la couche d'auto-attention et conjointement au niveau des clés et des valeurs, avec des clés et des valeurs fortement similaires entre elles au sein d'une phrase sur la dernière couche des modèles. On note aussi qu'elle se construit progressivement, les modèles avec plus de couches permettant une convergence plus forte. Rappelons que : (a) les clés impactent en premier lieu l'attention, des clés rassemblées dans un cône étroit résultant dans une attention distribuée sur l'ensemble des *tokens* de l'énoncé ; (b) les valeurs impactent les plongements en sortie qui sont définis comme une combinaison convexe de l'ensemble des valeurs.

En résumé, dans sa quête d'alignement des plongements dans une direction correspond à une classe, la couche d'auto-attention s'appuie naturellement sur les valeurs pour assurer cette convergence. Hormis pour e-SNLI, cela s'accompagne d'une convergence des clés qui explique les observations de plusieurs études sur la distribution uniforme de l'attention.

5. Par construction des clés et des valeurs, ce cas ne peut pas arriver en théorie avec des matrices de clés ou de valeurs.

4.2 Une tendance à compenser la régularisation de l'attention

Les observations précédentes nous amènent à nous interroger sur le contrôle de l'attention – par exemple pour offrir des capacités d'explication en forçant la parcimonie ou dans un cadre d'apprentissage guidé dans le cadre des approches en *rationalized learning* – et sur son impact sur la convergence des plongements *via* celle des clés et des valeurs.

Ce que nous avons observé dans les section précédentes montre que la direction prise par le plongement du *token* [CLS] est discriminante pour la tâche de classification. Cette direction est portée par les valeurs v_j pour lesquels le poids d'attention a_{0j} (0 étant l'indice du *token* [CLS]) est grand devant les autres. Dans la pratique, nous avons vu que les poids d'attention ont tendance à être uniformes, le *token* [CLS] étant alors le barycentre des valeurs v_j. En régularisant l'attention pour la rendre plus parcimonieuse, c'est-à-dire en supprimant l'uniformité des poids d'attention, nous changeons cependant la direction prise par le plongement du vecteur [CLS]. Or des études antérieurs montrent que cette opération ne change en rien la performance en classification du modèle, ce qui semble indiquer à son tour que cette régularisation n'empêche pas la convergence des plongements. Notre hypothèse est que pour compenser les contraintes imposées aux poids d'attention tout en garantissant la convergence, le modèle joue sur les valeurs en en renforçant la convergence dans un cône étroit pour une phrase donnée ; ainsi, toute modification de l'attention aura un impact très limité sur le plongement résultant de la combinaison des valeurs. Une alternative réside dans l'orthogonalisation des clés afin de concentrer sur quelques *tokens* la réponse à une requête donnée.

Pour vérifier cette hypothèse, nous reprenons les expériences de régularisation introduites par Nguyen *et al.* (2022) où la fonction de coût à minimiser à l'apprentissage inclut une minimisation de l'entropie des poids d'attention en plus de la minimisation de l'erreur de classification. Un paramètre λ permet de contrôler l'importance de la régularisation des poids d'attention par le critère d'entropie : plus λ est élevé, plus les poids d'attention se concentrent sur un petit nombre de *tokens*. Nous appliquons ici cette approche sur un modèle avec un seul bloc d'auto-attention.

Le tableau 4 rapporte la similarité moyenne des clés et des valeurs ainsi que l'entropie moyenne des poids d'attention pour différentes valeurs de λ. Ces mesures mettent clairement en évidence que lorsque la régularisation prend de l'importance, la concentration des poids d'attention (mesurée par l'entropie) augmente et l'alignement dans un cône des valeurs est renforcé pour compenser et garantir la convergence des plongements finaux dans une direction dépendante de la classe. Parallèlement, ces résultats réfutent l'hypothèse d'orthogonalisation des clés pour garantir la parcimonie de l'attention. Nous avons observé que le modèle joue avant tout sur la norme des clés pour rendre l'attention parcimonieuse plutôt que sur la direction. Cette dernière remarque illustre remarquablement la capacité d'une couche d'auto-attention à tirer parti de ces différents paramètres pour compenser les contraintes qui lui sont imposées et permettre de remplir sa fonction première d'alignement des plongements dans une direction dépendante de la classe.

5 Discussion

Les résultats présentés dans cette étude éclairent plusieurs points sur le fonctionnement des modèles transformeurs et, en particulier, le fonctionnement de la couche d'auto-attention dans les tâches de classification d'énoncé. L'ensemble des observations indique que la concentration des plongements d'un énoncé dans une direction, phénomène observé dans plusieurs études, est inhérente au fonction-

		$\lambda = 0$	$\lambda = 0.0001$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$
Hatexplain	*H*	0.988	0.31	0.111	0.021	0.01
	Key	0.745	0.669	0.687	0.732	0.727
	Value	**0.578**	**0.637**	**0.782**	**0.828**	**0.772**
E-SNLI	*H*	0.999	0.576	0.64	0.000	0.000
	Key	0.674	0.737	0.795	0.833	0.707
	Value	**0.507**	**0.45**	**0.526**	**0.868**	**0.925**
YelpHat	*H*	0.47	0.48	0.57	0.49	0.07
	Key	0.631	0.657	0.634	0.607	0.672
	Value	**0.303**	**0.391**	**0.375**	**0.366**	**0.483**

TABLE 4 – Entropie moyenne des poids d'attention (H) et similarité intra-énoncé moyenne des clés (K) et des valeurs (V) pour un modèle à 1 couche en fonction de la régularisation des poids d'attention.

nement du mécanisme d'auto-attention qui, pour la tâche de classification, cherche à « envoyer » la représentation de l'énoncé dans une direction qui dépend de la classe. Pour ce faire, le modèle utilise tous les degrés de liberté qui lui sont offerts en mettant en œuvre un jeu entre clés et valeurs : il en résulte une concentration dans une direction des clés et des valeurs. Ces dernières sont en première ligne pour garantir l'alignement des plongements, le modèle compensant les éventuelles contraintes sur les clés (ou sur l'attention) grâce aux valeurs et en jouant sur la norme des clés.

Par certains côtés, ces observations éclairent le débat sur l'utilisation de l'auto-attention pour expliquer une décision en mettant en avant les *tokens* importants dans la décision. Dans un contexte de classification d'énoncé, cette approche semble vouée à l'échec puisque le modèle cherchera forcément à construire sa convergence des plongements vers un cône donné, résultant inexorablement en une attention uniformément distribuée. La construction progressive de la convergence au travers des couches montre également que si une attention doit être utilisée pour expliquer une décision, elle est à considérer dans les premières couches du modèle avant que la convergence ne devienne trop forte.

Enfin, cet éclairage soulève de nombreuses questions sur la manière dont le modèle définit les directions correspondant à chacune des classes. Par exemple, est-ce que l'on a un comportement similaire pour les tâches d'étiquetage avec une direction par étiquette possible ? Est-ce qu'il existe un lien entre la dimension minimale/optimale de l'espace de plongement et le nombre de classe. Si l'on reste dans un contexte de classification d'énoncé, on peut également se demander si certains *tokens* sont responsables de la convergence dans une direction ou une autre, ce qui reviendrait à dire que, *in fine*, le modèle apprend les mots-clés, éventuellement dans leur contexte, qui permettent une bonne performance en classification. Se pose alors la question de comment mettre en évidence de tels mots-clés ou, de manière plus générale, celle de l'exploitation des propriétés géométriques que nous avons mises en évidence pour assurer une explication de la décision de classification.

Références

BAI B., LIANG J., ZHANG G., LI H., BAI K. & WANG F. (2021). Why attentions may not be interpretable ? In *27th Conference on Knowledge Discovery & Data Mining*, p. 25–34.

BIBAL A., CARDON R., ALFTER D., WILKENS R., WANG X., FRANÇOIS T. & WATRIN P. (2022). Is attention explanation ? An introduction to the debate. In *60th Annual Meeting of the Association for Computational Linguistics*, p. 3889–3900.

BOWMAN S. R., ANGELI G., POTTS C. & MANNING C. D. (2015). A large annotated corpus for learning natural language inference. In *2015 Conference on Empirical Methods in Natural Language Processing*, p. 632–642.

BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, p. 1877–1901.

CAI X., HUANG J., BIAN Y. & CHURCH K. (2021). Isotropy in the contextual embedding space : Clusters and manifolds. In *9th International Conference on Learning Representations*.

CAMBURU O.-M., ROCKTÄSCHEL T., LUKASIEWICZ T. & BLUNSOM P. (2018). e-SNLI : Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31, p. 9539–9549.

CARTON S., KANORIA S. & TAN C. (2022). What to learn, and how : Toward effective learning from rationales. In *Findings of the Association for Computational Linguistics*, p. 1075–1088.

CLARK K., KHANDELWAL U., LEVY O. & MANNING C. D. (2019). What does BERT look at ? An analysis of BERT's attention. In *2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 276–286.

CONNEAU A. & LAMPLE G. (2019). Cross-lingual language model pretraining. In *Advances in neural information processing systems*, volume 32, p. 7027–7037.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *17th Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 4171–4186.

DEYOUNG J., JAIN S., RAJANI N. F., LEHMAN E., XIONG C., SOCHER R. & WALLACE B. C. (2020). ERASER : A benchmark to evaluate rationalized nlp models. In *Annual Meeting Association for Computational Linguistics*, p. 4443–4458.

ETHAYARAJH K. (2019). How contextual are contextualized word representations ? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 55–65.

FOSSE L., NGUYEN D., SÉBILLOT P. & GRAVIER G. (2022). Une étude statistique des plongements dans les modèles transformers pour le français. In *29th Conference Traitement Automatique des Langues Naturelles*, p. 247–256.

HERNANDEZ E. & ANDREAS J. (2021). The low-dimensional linear geometry of contextualized word representations. In *25th Conference on Computational Natural Language Learning*, p. 82–93.

HTUT P. M., PHANG J., BORDIA S. & BOWMAN S. R. (2019). Do attention heads in BERT track syntactic dependencies ? Unpublished arXiv preprint 1911.12246.

JAIN S. & WALLACE B. C. (2019). Attention is not explanation. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, volume 1, p. 3543–3556.

JAWAHAR G., SAGOT B. & SEDDAH D. (2019). What does BERT learn about the structure of language ? In *Annual Meeting of the Association for Computational Linguistics*, p. 3651–3657.

KOVALEVA O., ROMANOV A., ROGERS A. & RUMSHISKY A. (2019). Revealing the dark secrets of BERT. In *2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 4356–4365.

LIN Y., TAN Y. C. & FRANK R. (2019). Open Sesame : Getting inside BERT's linguistic knowledge. In *2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 241–253.

MAAS A. L., DALY R. E., PHAM P. T., HUANG D., NG A. Y. & POTTS C. (2011). Learning word vectors for sentiment analysis. In *49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 142–150.

MATHEW B., SAHA P., YIMAM S. M., BIEMANN C., GOYAL P. & MUKHERJEE A. (2021). HateXplain : A benchmark dataset for explainable hate speech detection. In *35th AAAI Conference on Artificial Intelligence*, p. 14867–14875.

MCGUIRE E. S. & TOMURO N. (2021). Sentiment analysis with cognitive attention supervision. In *34th Canadian Conference on Artificial Intelligence*.

MOHANKUMAR A. K., NEMA P., NARASIMHAN S., KHAPRA M. M., SRINIVASAN B. V. & RAVINDRAN B. (2020). Towards transparent and explainable attention models. In *58th Annual Meeting of the Association for Computational Linguistics*, p. 4206–4216.

NGUYEN D., GRAVIER G. & SÉBILLOT P. (2021). A study of the plausibility of attention between rnn encoders in natural language inference. In *20th IEEE Intl. Conf. on Machine Learning and Applications*, p. 1–7.

NGUYEN D., GRAVIER G. & SÉBILLOT P. (2022). Filtrage et régularisation pour améliorer la plausibilité des poids d'attention dans la tâche d'inférence en langue naturelle. In *29th Conference Traitement Automatique des Langues Naturelles*, p. 95–103.

NGUYEN M. & NGUYEN T. H. (2018). Who is killed by police : Introducing supervised attention for hierarchical LSTMs. In *27th International Conference on Computational Linguistics*, p. 2277–2287.

NICULAE V. & BLONDEL M. (2017). A regularized framework for sparse and structured neural attention. In *31st International Conference on Neural Information Processing Systems*, p. 3340–3350.

PARANJAPE B., JOSHI M., THICKSTUN J., HAJISHIRZI H. & ZETTLEMOYER L. (2020). An information bottleneck approach for controlling conciseness in rationale extraction. In *2020 Conference on Empirical Methods in Natural Language Processing*, p. 1938–1952.

PRUTHI D., GUPTA M., DHINGRA B., NEUBIG G. & LIPTON Z. C. (2020). Learning to deceive with attention-based explanations. In *58th Annual Meeting of the Association for Computational Linguistics*, p. 4782–4793.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). *Improving language understanding by generative pre-training*. Rapport interne, OpenAI.

REIF E., YUAN A., WATTENBERG M., VIEGAS F., COENEN A., PEARCE A. & KIM B. (2019). Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems*, volume 32, p. 8592–8600.

ROGERS A., KOVALEVA O. & RUMSHISKY A. (2020). A primer in bertology : What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, **8**, 842–866.

SASAKI S., HEINZERLING B., SUZUKI J. & INUI K. (2023). Examining the effect of whitening on static and contextualized word embeddings. *Information Processing Management*, **60**(3).

SEN C., HARTVIGSEN T., YIN B., KONG X. & RUNDENSTEINER E. (2020). Human attention maps for text classification : Do humans and neural networks focus on the same words ? In *60th Annual Meeting of the Association for Computational Linguistics*, p. 4596–4608.

VAN AKEN B., WINTER B., LÖSER A. & GERS F. A. (2019). How does BERT answer questions ? a layer-wise analysis of transformer representations. In *28th ACM International Conference on Information and Knowledge Management*, p. 1823–1832.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, p. 6000–6010.

VOITA E., TALBOT D., MOISEEV F., SENNRICH R. & TITOV I. (2019). Analyzing multi-head self-attention : Specialized heads do the heavy lifting, the rest can be pruned. In *57th Annual Meeting of the Association for Computational Linguistics*, p. 5797–5808.

WIEGREFFE S. & PINTER Y. (2019). Attention is not not explanation. In *2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 11–20.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Un traitement hybride du vague textuel : du système expert VAGO à son clone neuronal

Benjamin Icard[(1)], Vincent Claveau[(2)], Ghislain Atemezing[(3)], Paul Égré[(1)]

(1) Institut Jean-Nicod, PSL University, 29 rue d'Ulm, 75005 Paris, France
(2) CNRS, IRISA, 263 Av. Général Leclerc, 35000 Rennes, France
(3) Mondeca, 18 rue de Londres, 75009 Paris, France

RÉSUMÉ

L'outil VAGO est un système expert de détection du vague lexical qui mesure aussi le degré de subjectivité du discours, ainsi que son niveau de détail. Dans cet article, nous construisons un clone neuronal de VAGO, fondé sur une architecture de type BERT, entraîné à partir des scores du VAGO symbolique sur un corpus de presse française (FreSaDa). L'analyse qualitative et quantitative montre la fidélité de la version neuronale. En exploitant des outils d'explicabilité (LIME), nous montrons ensuite l'intérêt de cette version neuronale d'une part pour l'enrichissement des lexiques de la version symbolique, et d'autre part pour la production de versions dans d'autres langues.

ABSTRACT

A hybrid treatment of textual vagueness: enriching the expert system VAGO with a neural clone

The VAGO tool is an expert system for lexical vagueness detection that also measures the degree of subjectivity of the speech, as well as its level of detail. In this paper, we build a neural clone of VAGO, based on a BERT-like architecture, trained on symbolic VAGO scores on a French press corpus (FreSaDa). The qualitative and quantitative analysis shows the fidelity of the neural version. By exploiting explainability tools (LIME), we then show the interest of this neural version for the enrichment of the lexicons of the symbolic version, and for the production of versions in other languages.

MOTS-CLÉS : Vague - Subjectivité - Précision - Détail - Hybridation - Explicabilité.

KEYWORDS: Vagueness - Subjectivity - Precision - Detail - Hybridization - Explainability.

1 Introduction

L'évaluation de la qualité d'un texte ou d'un discours s'avère nécessaire dans beaucoup d'applications, lesquelles reposent alors sur une définition propre de la notion de *qualité* (Štajner *et al.*, 2022). Pour un apprenant, on peut par exemple chercher à mesurer la complexité lexicale (Shardlow *et al.*, 2021) ou syntaxique des énoncés (Chen & Zechner, 2011). On peut également mesurer la complexité conceptuelle des termes et des énoncés à destination d'un jeune public (Štajner & Hulpuş, 2018) ou la compréhensibilité de formules administratives (François *et al.*, 2014). La lisibilité est une autre dimension largement explorée à partir d'indices divers (Collins-Thompson, 2014), allant jusqu'au rendu graphique pour certains troubles de la lecture (Rello & Baeza-Yates, 2016). S'agissant de la cohérence, on peut s'intéresser à l'enchaînement logique des énoncés, vu alors comme une tâche

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 151–163.

d'*entailment* (Poliak, 2020). Notons enfin que ces questions de qualité de texte se posent également pour l'évaluation de systèmes de TAL produisant du texte (Celikyilmaz *et al.*, 2020), comme le résumé automatique, la traduction, la génération de texte, la simplification, etc.

Dans cet article, nous nous intéressons à une autre dimension de qualité du discours : la notion d'*informativité*. Là encore, des travaux existants explorent cette notion sous différents prismes. Tewari *et al.* (2020) utilisent par exemple des critères de cohésion syntaxique, alors que d'autres exploitent des critères de nouveauté par rapport à un ensemble de connaissances préexistantes (Chen *et al.*, 2018; Shibayama *et al.*, 2021). Pour notre part, nous nous inscrivons dans une veine de travaux où la mesure du degré d'informativité consiste à évaluer le caractère plus ou moins précis ou vague du discours (Van Deemter, 2010; Égré, 2018). Plus un discours est précis, plus il est apte à être infirmé s'il est faux (Popper, 1963) ; inversement plus un discours est vague, moins il se prête à la réfutation empirique et plus il est susceptible de véhiculer des informations subjectives (Égré & Icard, 2018). Pour automatiser l'évaluation de la qualité informationnelle d'un texte, il est donc utile de détecter des indices de vague comme de précision.

Dans ce but, nous proposons ici de combiner deux approches. D'une part, nous faisons appel à un système expert de détection et de mesure du vague lexical, l'outil VAGO (Guélorget *et al.*, 2021; Icard *et al.*, 2022). De l'autre, nous proposons d'en créer une version neuronale afin de tester comme d'enrichir les performances du système expert. L'un des enjeux de cette méthode est d'étendre à d'autres langues que le français et l'anglais les résultats du système expert. Un autre est d'avancer dans la maîtrise de méthodes hybrides de traitement du langage.

2 L'outil symbolique VAGO

2.1 Typologie du vague et mesure du niveau de détail

VAGO mesure le vague et la subjectivité des documents à partir d'une base de données lexicales en français et en anglais (Atemezing *et al.*, 2021). Fondée sur une typologie issue de (Égré & Icard, 2018), cette base de données propose un inventaire des termes vagues en quatre catégories : vague d'approximation (V_A), vague de généralité (V_G), vague de degré (V_D) et vague combinatoire (V_C).

Le vague d'approximation concerne principalement des modificateurs comme *"environ"*, qui rendent moins strictes les conditions de vérité de l'expression qu'ils modifient. Le vague de généralité comprend des déterminants comme *"certains"*, ainsi que des modificateurs comme *"au plus"*. Contrairement aux expressions d'approximation, ces dernières ont des conditions de vérité précises. La classe des expressions relevant du vague de degré et du vague combinatoire (Alston 1964) comprend principalement des adjectifs unidimensionnels d'une part (tels que *"grand"*, *"vieux"*) et des adjectifs multidimensionnels d'autre part (comme *"beau"*, *"intelligent"*, *"bon"*, *"qualifié"*). Les expressions de type V_A et V_G sont en outre traitées comme des expressions *factuelles*, et les expressions de type V_D et V_C comme des expressions *subjectives* (Kennedy, 2013; Verheyen *et al.*, 2018; Solt, 2018).

Selon les règles de calcul de ratio détaillées en 2.2, dans la version princeps de VAGO il suffit qu'une phrase contienne au moins un marqueur de vague, respectivement de vague subjectif, pour que VAGO la considère comme vague, respectivement comme subjective[1]. Cependant, cette version n'offre pas

[1]En plus de ces différents ratios, VAGO s'appuie sur plusieurs règles de modulation des scores de vague en fonction du

de mesure *positive* pour la précision du discours: une phrase est jugée précise si elle ne contient aucun marqueur vague ; un texte est jugé précis s'il ne contient aucune phrase vague. À titre d'exemple, la version en ligne de VAGO princeps attribuera des scores de vague et de subjectivité identiques, égaux à 1, aux deux phrases suivantes[2]:

(a) "Roi de Naples de 1806 à 1808, puis d'Espagne de 1808-1813, il est un personnage **important** du dispositif que met en place Napoléon pour asseoir la souveraineté de la France sur l'Europe continentale".

(b) "Pour guérir **rapidement** de la Covid-19 il faut prendre une **excellente** décoction de plantes".

Les phrases (a) et (b) contiennent chacune au moins un marqueur de vague également vecteur de subjectivité: un seul pour (a) avec "*important*" et deux pour (b) avec "*rapidement*" et "*excellente*". Ces phrases sont donc identiquement vagues/subjectives selon VAGO princeps. Intuitivement, pourtant, la phrase (a) qui contient neuf entités nommées (termes soulignés dans la phrase) est plus informative que la phrase (b) qui ne contient qu'une seule entité nommée ("*Covid-19*") et renferme donc moins de détails que (a). Pour combler cette lacune, la version princeps de VAGO est enrichie ici d'un score de détail fondée sur la part relative des entités nommées comparées aux expressions vague.

2.2 Scores VAGO : vague, subjectivité, détail

Actuellement, VAGO permet de mesurer le score de vague, de subjectivité et le niveau de détail de documents anglais ou français. La détection du vague et de la subjectivité s'appuie sur la base de donnée princeps comportant actuellement 1 640 termes dans les deux langues (Atemezing *et al.*, 2022), répartis comme suit par catégorie de vague : $|V_A| = 9$, $|V_G| = 18$, $|V_D| = 43$ et $|V_C| = 1570$. Dans le cas du niveau de détail, la détection se fonde sur l'identification des entités nommées par spaCy[3] (personnes, localités, indications temporelles, institutions, nombres). Pour chaque mesure, les marqueurs caractéristiques sont détectés et notés des mots vers les phrases, puis des phrases vers les textes.

Pour une phrase ϕ donnée, son *score de vague* est défini comme le rapport entre le nombre de mots vagues dans ϕ et le nombre total de mots dans la phrase N_ϕ :

$$R_{vague}(\phi) = \frac{\overbrace{|V_D|_\phi + |V_C|_\phi}^{subjectif} + \overbrace{|V_A|_\phi + |V_G|_\phi}^{factuel}}{N_\phi} \quad (1)$$

où $|V_A|_\phi$, $|V_G|_\phi$, $|V_D|_\phi$ et $|V_C|_\phi$ représentent le nombre de termes dans ϕ relevant de chacune des quatre catégories de vague (approximation, généralité, vague de degré et vague combinatoire). Plus finement, le *score de subjectivité* d'une phrase est calculé comme le rapport entre les expressions vagues subjectives et le nombre total de mots de la phrase. On peut calculer un score de vague factuel

contexte (voir Icard *et al.*, 2022).

[2]La phase (a) est tirée de l'article Wikipédia sur Joseph Bonaparte, tandis que la phrase (b) est inspirée d'une fausse nouvelle ou "*fake news*".

[3]https://spacy.io/

identiquement avec les expressions de généralité et d'approximation (cf. sections 3 et 4) :

$$R_{subjectif}(\phi) = \frac{|V_D|_\phi + |V_C|_\phi}{N_\phi} \qquad R_{vaguefactuel}(\phi) = \frac{|V_A|_\phi + |V_G|_\phi}{N_\phi} \tag{2}$$

Le *score de détail* d'une phrase peut être défini comme le ratio $R_{dtail}(\phi) = \frac{|P|_\phi}{N_\phi}$, où $|P|_\phi$ désigne le nombre d'entités nommées de la phrase (termes référentiels). Par extension, si $|V|_\phi$ désigne le nombre de termes vagues d'une phrase (toutes catégories confondues), on définit le *score de détail/vague* d'une phrase comme la part relative des entités nommées, soit :

$$R_{détail/vague}(\phi) = \frac{|P|_\phi}{|P|_\phi + |V|_\phi} \tag{3}$$

Dans l'exemple précédent, on peut vérifier que $R_{détail/vague}(\text{a}) = 9/10$, alors que $R_{détail/vague}(\text{b}) = 1/3$, ce qui donne une meilleure mesure du caractère plus informatif de (a).

Pour des ensembles de phrases, ou textes T, les scores de vague de T (respectivement de vague subjectif, ou de vague factuel) sont définis comme la proportion de phrases de T dont le score de vague (resp. subjectif, ou factuel) sont non nuls. Le score de détail/vague de T, lui, est défini comme la moyenne des ratios $R_{détail/vague}$ de chacune des phrases de T.

2.3 Implémentation des mesures

En terme d'ingénierie, VAGO s'appuie sur GATE (Cunningham, 2002) pour le traitement des corpus. L'algorithme exploite également l'annotateur de contenu sémantique CA-Manager (Cherfi *et al.*, 2013) qui sert à extraire des connaissances à partir de données non structurées. En l'état, VAGO détecte automatiquement la langue du corpus (anglais ou français) à l'aide du module TextCat[4]. La version neuronale de VAGO présentée en section 3 utilise spaCy pour la détection des entités nommées.

L'outil VAGO en ligne, disponible sur le site de Mondeca[5], présente les fonctionnalités de VAGO dans sa version princeps. Le site propose une interface graphique pour mesurer les scores de vague et de subjectivité de textes sous la forme de deux baromètres. Le premier baromètre représente le degré de vague d'un texte (défini comme $R_{vague}(T)$) tandis que le second baromètre indique le degré auquel le texte rapporte une opinion plutôt qu'un fait, autrement dit la proportion de vocabulaire subjectif au sein du texte (défini comme $R_{subjectif}(T)$).

2.4 Test de VAGO sur la presse française

VAGO a été testé sur le corpus français "FreSaDa"[6] (Ionescu & Chifu, 2021) composé de 11 570 articles de presse répartis en deux classes supposées homogènes : 5 648 articles "réguliers" et provenant de la presse française généraliste, sans faire de présupposition sur leur vérité ou leur

[4] https://www.let.rug.nl/vannoord/TextCat/index.html
[5] https://research.mondeca.com/demo/vago/
[6] https://github.com/adrianchifu/FreSaDa

fausseté, versus 5 922 articles “satiriques” et explicitement faux à ce titre. Au sein du corpus total, VAGO a traité 10 969 articles des 11 570 articles initiaux, — les 601 articles restants ayant été écartés car ils ne relèvent pas d’un format adéquat pour être traité par l’outil (mots isolés, mots-clés, phrases incomplètes, etc.). Les résultats fournis par VAGO sont rapportés en Figure 1.

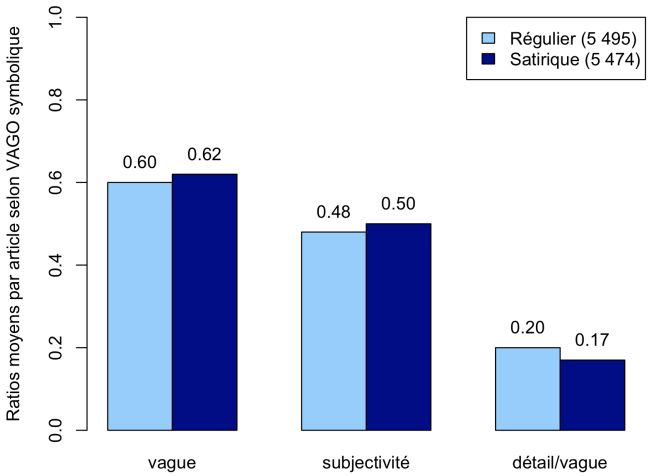


Figure 1: Ratios moyens par article du corpus FreSaDa selon VAGO.

Selon VAGO, les articles du corpus satirique sont significativement *plus vagues* ($p = 4.99 \times 10^{-11}$), *plus subjectifs* ($p = 1.69 \times 10^{-9}$) et *moins détaillés* ($p = 3.36 \times 10^{-22}$) que les articles du corpus de presse régulier (scores calculés par textes ; t-tests bilatéraux, $\alpha = 0.05$, avec correction de Bonferroni). Ces résultats sont conformes aux attentes et corroborent les résultats obtenus antérieurement avec VAGO sur des textes en anglais (Guélorget *et al.*, 2021). En outre, des expériences non rapportées dans cet article montrent que les ratios calculés par VAGO, utilisés en entrée d’un classifieur, permettent de distinguer les documents selon leur catégorie *légitime* ou *biaisée* avec une grande précision.

3 La version neuronale VAGO-N

À partir d’une architecture BERT (Devlin *et al.*, 2018), nous présentons ici une version neuronale VAGO-N de la version symbolique VAGO décrite précédemment. Cette version neuronale vise à dépasser certaines limites de VAGO et à ouvrir la voie à plusieurs développements que nous présentons dans la section suivante.

3.1 Apprentissage du clone VAGO-N

L’architecture BERT est associée à une couche de régression ainsi qu’à une fonction de perte MSE afin de prédire un score associé à un texte ; nous testons les deux type de vague : subjectif ou factuel. Par souci de complétude, nous testons également la prédiction du score $R_{détail}$, mais ce score peut être plus simplement calculé à partir d’un système de reconnaissance d’entités nommées ; nous n’y revenons donc pas dans les expériences suivantes. Comme pour une tâche de distillation, VAGO est donc utilisé pour associer un score de vague aux phrases d’un corpus et entraîner ainsi un système neuronal.

Dans les expériences rapportées, 106 000 phrases sont tirées aléatoirement des 10 969 articles du corpus FreSaDa traités par VAGO, et sont classiquement divisées en jeu d'entraînement (85 000 phrases) et de test (21 000 phrases). Nous utilisons un modèle RoBERTa Large (*Batch Size*=30 ; *Learning Rate*=1e-6 ; *Epochs*=20) ; des expériences non détaillées ici avec un modèle CamemBERT (Martin *et al.*, 2019) fournissent des résultats légèrement inférieurs.

Les performances sont rapportées en Table 1 avec les mesures standard de régression : l'erreur quadratique moyenne (RMSE), le coefficient de détermination (R^2), l'erreur absolue moyenne (MAE) et l'erreur absolue médiane (MedAE). Toutes ces mesures montrent que VAGO−N réplique avec une grande précision les scores du VAGO symbolique. La tâche de détection de la subjectivité semble un peu plus difficile que celle du vague factuel.

	RMSE	R^2	MAE	MedAE
vague subjectif	0.022063	0.859897	0.014518	0.009488
vague factuel	0.008745	0.949339	0.004124	0.001730
détail/vague	0.097008	0.882543	0.051396	0.012367

Table 1: Résultats de régression de VAGO−N sur les phrases du corpus FreSaDa (français) pour les scores de vague subjectif, de vague factuel et de détail par rapport au vague.

3.2 Comparaison des versions de VAGO

L'évaluation quantitative précédente indique que VAGO−N réplique assez fidèlement le comportement général de VAGO. Il est intéressant de vérifier de manière plus qualitative que cette version neuronale s'appuie bien sur les mêmes indices lexicaux que la version symbolique.

Pour cela, nous utilisons l'outil d'explicabilité LIME (Ribeiro *et al.*, 2016). Appliqué aux sorties de VAGO−N, LIME permet d'identifier les tokens qui contribuent le plus (ou le moins) au score de vague d'un texte donné. Dans le cas d'une phrase en français, un exemple de sortie de LIME concernant le cas du vague subjectif est fourni en Figure 2. Avec cet outil, nous examinons les cas où les prédictions (scores de vague) de VAGO−N divergent le plus de celles de VAGO.

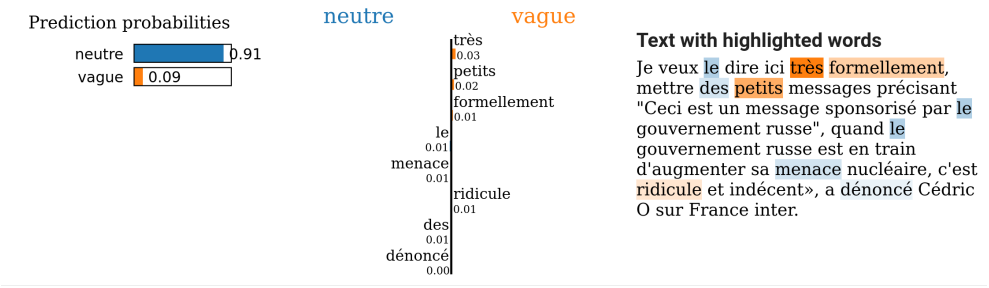


Figure 2: Exemple de sortie de LIME sur une phrase du corpus FreSaDa traitée par VAGO−N. La catégorie notée neutre correspond à la catégorie inverse du vague, contribuant négativement au score de vague.

L'étude de ces cas d'erreurs fait ressortir plusieurs points. Concernant les écarts de prédiction entre VAGO et VAGO−N sur le vague factuel, la très grande majorité des termes identifiés comme contribuant

le plus fortement à la prédiction de VAGO−N figurent déjà au sein du lexique français de VAGO, aussi bien pour le vague de généralité V_G (e.g. *"tout/tous/toutes", "jamais", "ou", "général", "quelques", "certains/certaines"*), que pour le vague d'approximation (*"environ", "presque"*). Les indices de vague factuel repérés sont corrects mais leur poids dans le score final de VAGO−N diffère du calcul effectué par le VAGO initial qui n'établit pas de pondération. Cette différence de poids attribuée par VAGO−N peut résulter de mots d'autres catégories morpho-syntaxiques (les lexiques de VAGO se focalisant sur les adjectifs et adverbes) qui viennent amplifier ou amoindrir le score de vague factuel résultant.

De manière similaire, dans le cas du vague subjectif, LIME appliqué à VAGO−N relève des adjectifs et des termes d'outrance déjà présents dans le lexique existant, soit au sein du vague combinatoire pour majorité (e.g. *"négatif", "affirmatif", "intéressant", "fortement", "difficile", "probablement", "vrai", "stupide", "vraiment"*), soit au sein du vague de degré (*"petit"*). LIME identifie également d'autres adjectifs porteurs de vague combinatoire qui ne figurent pas encore dans le lexique mais sont voués à y figurer (e.g. *"durable", "particulièrement", "ringard", "actuel"*), avec toutefois des exceptions ("*sabbatique*") ; nous y revenons dans la section suivante.

4 Extensions des approches VAGO

La section précédente montre qu'il est possible de construire un équivalent de la version française de VAGO par apprentissage. Cela permet d'explorer plusieurs développements que nous présentons dans les deux sous-sections suivantes : l'enrichissement de la base lexicale au coeur de VAGO, et la construction de systèmes de détection du vague pour d'autres langues.

4.1 Validation et enrichissement du VAGO symbolique

Comme nous l'avons vu précédemment, pour chaque token t dans un texte, LIME fournit un score de contribution de t à la prédiction de vague (subjectif ou factuel) du texte que nous notons $c_{occ}(t)$. Dans le cas d'une phrase, plus un terme t reçoit un score $c_{occ}(t)$ élevé, plus il contribue positivement au score de vague de cette phrase.

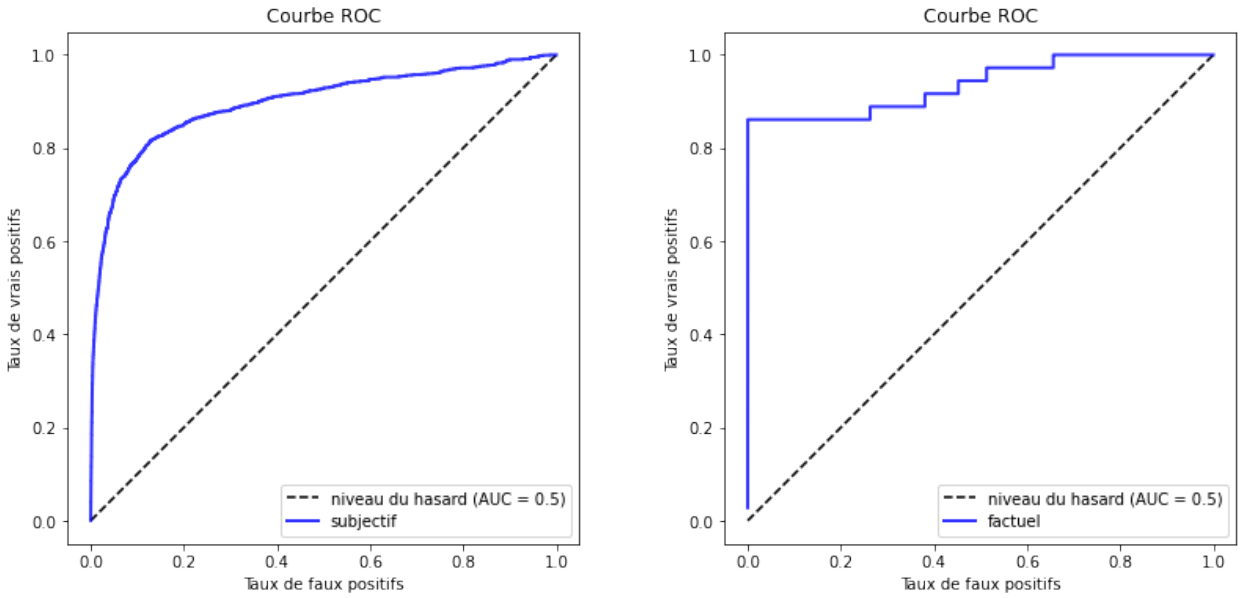
En appliquant LIME aux phrases des 10 969 articles du corpus FreSaDa traités par VAGO et exploités dans VAGO−N, nous collectons les scores de contribution c_{occ} de toutes les occurrences de tous les tokens figurant au sein de ces phrases. Pour obtenir un score global $c_{tok}(t)$ par token t, nous sommons et normalisons les c_{occ} par le nombre total d'occurrences de chaque token noté ici $|occ_t|$: $c_{tok}(t) = \frac{1}{|occ_t|} \sum_{o \in occ_t} c_{occ}(o)$. Notre hypothèse est que des termes du lexique de VAGO devraient se retrouver prioritairement parmi les tokens recevant les c_{tok} les plus élevés. À cet égard, nous calculons les précisions statistiques P@i (vrais positifs/(vrais+faux positifs)) sur la liste des tokens ordonnées par c_{tok} décroissants. À noter qu'un token est pris en compte s'il est une flexion d'un terme du lexique VAGO. Les résultats sont recensés en Table 2.

La Figure 3 présente la courbe ROC reliant le score c_{tok} à la présence au sein du lexique VAGO. Ces résultats établissent le bien-fondé de notre hypothèse. Ainsi, VAGO−N, bien qu'entraîné uniquement sur des phrases et leur score, est capable de reconstruire le lexique au cœur de la version symbolique.

De plus, nous avons examiné les 100 tokens détenteurs des c_{tok} les plus élevés pour le vague

	P@5	P@10	P@20	P@30	P@100	P@200
vague subjectif	1.00	1.00	0.95	0.93	0.81	0.79
vague factuel	1.00	1.00	1.00	0.93	0.31	0.16

Table 2: Comparaison de la précision obtenue à différents seuils de la liste de tokens du lexique VAGO ordonnée par c_{tok}.



(a) vague subjectif

(b) vague factuel

Figure 3: Courbe ROC de c_{tok} comme indicateur de présence dans le lexique français de VAGO ; a) correspond au vague subjectif, b) au vague factuel.

subjectif : outre les 81 bien présents dans le lexique, quelques formes verbales sont listées. Bien que considérés comme des faux positifs (les lexiques VAGO actuels ne recensant que les adjectifs et adverbes), leur pertinence peut être discutée. Dans les tokens restants, sept mots absents ont été validés comme pertinents et méritant d'être intégrés dans les lexiques. Cette liste contient quatre adverbes (*"également", "seulement", "particulièrement", "clairement"*) pour lesquels la base de données VAGO contient les adjectifs racines dans deux cas (*"particulier", "clair"*) ; un verbe d'action (*"faire"*) ; un adjectif pouvant également être un nom (*"droit/droite"*) ; et un nom (*"nombre"*). On note également la détection de formes non standard de termes présents dans le lexique (*"difficille", "pauv"*), illustrant la robustesse de l'approche neuronale sur du texte bruité (coquille, abréviation...). Les résultats obtenus valident le clone neuronal VAGO-N qui retrouve les indices lexicaux du système expert VAGO tout en permettant d'en identifier de nouveaux ou des formes non standard.

4.2 Développement de VAGO-N multilingues

Le développement de versions symboliques de VAGO pour d'autres langues implique de disposer de lexiques de vague dans les langues cibles. Cependant, la traduction automatique de ces lexiques n'est pas possible en raison du caractère idiomatique et hors contexte des listes d'expressions en jeu.

	RMSE	R^2	MAE	MedAE
vague subjectif	0.031801	0.708915	0.022865	0.016807
vague factuel	0.016990	0.808772	0.009582	0.004172

Table 3: Résultats de régression de VAGO-N sur les phrases du corpus FreSaDa (traduites automatiquement en anglais) concernant les scores de vague subjectif et de vague factuel.

	P@5	P@10	P@20	P@30	P@100	P@200
vague subjectif	0.80	0.90	0.90	0.93	0.90	0.84
vague factuel	1.00	0.80	0.55	0.50	0.26	0.14

Table 4: Comparaison de la précision obtenue à différents seuils de la liste de tokens ordonnée par c_{tok} en fonction du lexique VAGO.

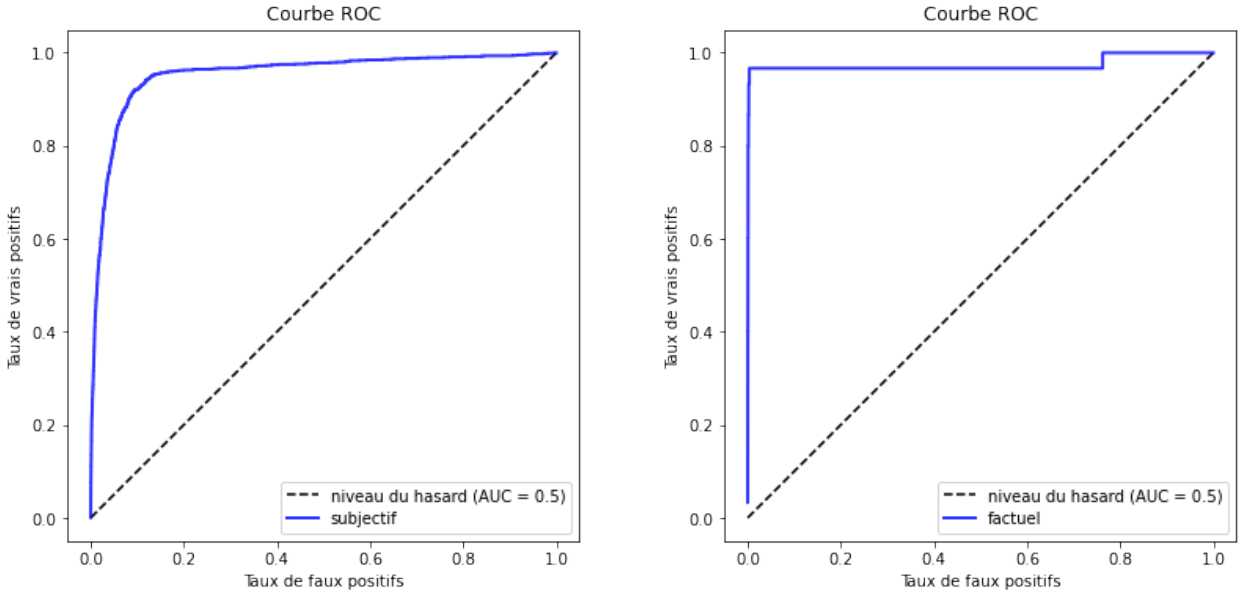
Cela étant, il est possible de traduire le jeu d'entraînement de VAGO-N en faisant l'hypothèse suivante : les scores de vague, en particulier de vague subjectif et de vague factuel, sont conservés de la langue source vers la langue cible. Pour commencer, le corpus FreSaDa a été traduit du français vers l'anglais en utilisant le modèle Helsinki-NLP/opus-mt-fr-en[7] (Tiedemann & Thottingal, 2020). Ensuite, VAGO-N a été entraîné à prédire les scores de vague subjectif et de vague factuel sur ce corpus en anglais (en utilisant les mêmes hyper-paramètres que pour l'entraînement de VAGO-N sur le français). Les résultats de régression sont similaires à ceux obtenus pour le français, et présentés en Table 3.

En appliquant la même approche qu'en sous-section 4.1, nous isolons la liste des tokens ordonnées par c_{tok} décroissants, puis la comparons au lexique anglais de VAGO servant ainsi de vérité-terrain. La précision de cette liste mesurée à différents seuils est rapportée en Table 4. Les courbes ROC correspondantes sont présentées en Figure 4.

Nous collectons également les 100 termes anglais les plus porteurs de vague selon VAGO-N. Ces termes sont comparés à ceux du lexique anglais de la version symbolique : 90 termes figurent déjà au sein du lexique anglais de VAGO.

Parmi les termes de rang le plus élevé parmi les 100 premiers qui ne figurent pas dans VAGO, on trouve cinq adjectifs ou adverbes ayant vocation à figurer dans le vague combinatoire (*"likely", "full", "complicated", "frankly", "enough"*), un modal (*"must"*), qu'il est également sensé d'intégrer étant donné la présence de *"should"* dans le lexique VAGO. Quatre termes en revanche ne relèvent pas clairement du vague (*"course", "lost", "lose" et "finally"*), sauf éventuellement le premier (occurrences de "of course" dont l'usage est subjectif). Parmi les 100 termes suivants, tous les termes qui ne figurent pas dans VAGO sont des adjectifs pouvant figurer dans la catégorie V_C (*"worse", "complex"*, etc.).

[7] https://huggingface.co/Helsinki-NLP/opus-mt-fr-en



(a) vague subjectif

(b) vague factuel

Figure 4: Courbe ROC de c_{tok} comme indicateur de présence dans le lexique anglais de VAGO ; a) correspond au vague subjectif, b) au vague factuel à partir de textes traduits en anglais.

5 Conclusion

Dans cet article, nous sommes partis d'un système expert de détection du vague, l'outil VAGO, qui associe à des textes des scores de vague, de subjectivité, et de niveau de détail/vague, et nous avons ensuite créé un clone neuronal de VAGO fondé sur BERT. À la différence de VAGO, VAGO−N est entraîné uniquement à partir des scores de VAGO, sans connaître le lexique sous-jacent à cette version symbolique. En utilisant LIME, il apparaît que les termes dont la contribution aux scores du VAGO−N sont les plus élevés sont ou bien des termes figurant dans VAGO, ou des termes ayant majoritairement vocation à y figurer. Ce résultat suggère que les décisions de VAGO−N s'expliquent dans une large mesure à partir des items lexicaux identifiés par le VAGO symbolique. On a pu monter l'intérêt de ces croisements entre système symbolique et neuronal : une fois appris, VAGO−N permet de compléter les lexiques du VAGO symbolique, il permet de produire facilement des versions neuronales dans d'autres langues, et rend possible des versions symboliques dans ces langues en produisant les lexiques nécessaires.

Beaucoup de pistes restent à explorer. Il serait notamment intéressant de comparer les indices lexicaux trouvés par LIME dans un classifieur entrainé à distinguer directement les types de documents de FreSada (ou d'autres corpus pour la détection de Fake News) et ceux obtenus par notre approche, que nous espérons plus génériques. Une autre piste que nous souhaitons explorer pour mesurer la part de généricité de notre approche serait de masquer les entités nommées dans un texte pour voir si les scores de VAGO−N restent stables avant et après ce masquage, et pour déterminer la part revenant proprement au lexique VAGO (et qui ne contient pas d'entités nommées) dans la décision du VAGO−N. Le système expert VAGO est par définition un système rigide qui ne différencie par le caractère plus ou moins vague d'un terme selon le contexte. Prenons un terme comme *"affirmatif"* : le fait pour une phrase d'être *"affirmative"* n'est pas vague, en revanche si une personne est dite *"très affirmative"*,

alors *"affirmatif"* revêt un sens vague et évaluatif. À la différence de VAGO, VAGO−N est capable de différencier les scores de contribution de ces différentes occurrences d'un terme. Parmi les questions que nous réservons à un examen ultérieur, il serait particulièrement fructueux d'examiner la variabilité des scores de termes vagues selon les différents contextes dans lesquels ils apparaissent. La question se pose plus généralement pour BERT, de savoir si les plongements des termes vagues manifestent une dispersion plus grande que les plongements des termes fonctionnels et précis du lexique.

Remerciements

Nous remercions deux rapporteurs pour leur commentaires, ainsi que Guillaume Gravier (CNRS – IRISA) pour son retour et ses suggestions sur la version préliminaire de cet article. Ce travail a été réalisé dans le cadre du programme HYBRINFOX (ANR-21-ASIA-0003) (CNRS, IRISA, Mondeca, Airbus). PE et BI remercient également le programme ANR-17-EURE-0017 (FrontCog), et PE le programme PLEXUS (Marie Skłodowska-Curie Action, Horizon Europe Research and Innovation Programme, grant agreement n°101086295).

6 Bibliographie

ALSTON W. P. (1964). *Philosophy of Language*. Prentice Hall.

ATEMEZING G., ICARD B. & ÉGRÉ P. (2021). Multilingual gazetteers to detect vagueness in textual documents. DOI : 10.5281/zenodo.4718530.

ATEMEZING G., ICARD B. & ÉGRÉ P. (2022). Vague Terms in SKOS to detect vagueness in textual documents. DOI : 10.5281/zenodo.4718530.

CELIKYILMAZ A., CLARK E. & GAO J. (2020). Evaluation of text generation: A survey. DOI : 10.48550/ARXIV.2006.14799.

CHEN D., MA S., YANG P. & SUN X. (2018). Identifying high-quality chinese news comments based on multi-target text matching model. DOI : 10.48550/ARXIV.1808.07191.

CHEN M. & ZECHNER K. (2011). Computing and evaluating syntactic complexity features for automated scoring of spontaneous non-native speech. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, p. 722–731, Portland, Oregon, USA: Association for Computational Linguistics.

CHERFI H., COSTE M. & AMARDEILH F. (2013). CA-manager: a middleware for mutual enrichment between information extraction systems and knowledge repositories. In *4th workshop SOS-DLWD*, p. 15–28.

COLLINS-THOMPSON K. (2014). Computational assessment of text readability: A survey of current and future research. *ITL - International Journal of Applied Linguistics*, **165**, 97–135. DOI : 10.1075/itl.165.2.01col.

CUNNINGHAM H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities*, **36**(2), 223–254. DOI : 10.1023/A:1014348124664.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*.

ÉGRÉ P. (2018). *Qu'est-ce que le vague?* Vrin.

ÉGRÉ P. & ICARD B. (2018). Lying and vagueness. In J. MEIBAUER, Éd., *Oxford Handbook of Lying*. OUP.

FRANÇOIS T., BROUWERS L., NAETS H. & FAIRON C. (2014). AMESURE: a readability formula for administrative texts (AMESURE: une plateforme de lisibilité pour les textes administratifs) [in French]. In *Proceedings of TALN 2014 (Volume 2: Short Papers)*, p. 467–472, Marseille, France: Association pour le Traitement Automatique des Langues.

GUÉLORGET P., ICARD B., GADEK G., GAHBICHE S., GATEPAILLE S., ATEMEZING G. & ÉGRÉ P. (2021). Combining vagueness detection with deep learning to identify fake news. In *Proceedings of 24th International Conference on Information Fusion*, p.8̃.

ICARD B., ATEMEZING G. & ÉGRÉ P. (2022). VAGO: un outil en ligne de mesure du vague et de la subjectivité. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle (PFIA 2022)*, p. 68–71.

IONESCU R.-T. & CHIFU A.-G. (2021). FreSaDa: A french satire data set for cross-domain satire detection. In *The International Joint Conference on Neural Network, IJCNN 2021*, IJCNN2021.

KENNEDY C. (2013). Two sources of subjectivity: Qualitative assessment and dimensional uncertainty. *Inquiry*, **56**(2-3), 258–277.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2019). CamemBERT: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

POLIAK A. (2020). A survey on recognizing textual entailment as an NLP evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, p. 92–109, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.eval4nlp-1.10.

POPPER K. R. (1963). *Conjectures and refutations: the growth of scientific knowledge*. New York: Basic Books.

RELLO L. & BAEZA-YATES R. (2016). The effect of font type on screen readability by people with dyslexia. *ACM Trans. Access. Comput.*, **8**(4). DOI : 10.1145/2897736.

RIBEIRO M. T., SINGH S. & GUESTRIN C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 1135–1144, New York, NY, USA: Association for Computing Machinery. DOI : 10.1145/2939672.2939778.

SHARDLOW M., EVANS R., PAETZOLD G. H. & ZAMPIERI M. (2021). SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, p. 1–16, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2021.semeval-1.1.

SHIBAYAMA S., YIN D. & MATSUMOTO K. (2021). Measuring novelty in science with word embedding. *PLoS ONE*, **16**(7). DOI : 10.1371/journal.pone.0254034.

SOLT S. (2018). Multidimensionality, subjectivity and scales: Experimental evidence. In E. CASTROVIEJO, L. MCNALLY & G. SASSOON, Éds., *The Semantics of Gradability, Vagueness, and Scale Structure*, p. 59–91. Springer.

ŠTAJNER S. & HULPUŞ I. (2018). Automatic assessment of conceptual text complexity using knowledge graphs. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 318–330, Santa Fe, New Mexico, USA: Association for Computational Linguistics.

ŠTAJNER S., SAGGION H., FERRÉS D., SHARDLOW M., SHEANG K. C., NORTH K., ZAMPIERI M. & XU W., Éds. (2022). *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

TEWARI M., BENSCH S., HELLSTRÖM T. & RICHTER K.-F. (2020). Modelling grice's maxim of quantity as informativeness for short text. In *:*, p. 1–7.

TIEDEMANN J. & THOTTINGAL S. (2020). OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

VAN DEEMTER K. (2010). *Not exactly: In praise of vagueness*. OUP Oxford.

VERHEYEN S., DEWIL S. & ÉGRÉ P. (2018). Subjectivity in gradable adjectives: The case of *tall* and *heavy*. *Mind & Language*, **33**(5), 460–479.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Uniformité de la densité informationnelle: le cas du redoublement du sujet

Yiming Liang[1] Pascal Amsili[2] Heather Burnett[1]
(1) Laboratoire de linguistique formelle, Université Paris Cité/CNRS, Paris, France
(2) Laboratoire Lattice, CNRS/PSL-ENS/Sorbonne Nouvelle, Paris, France
yiming.liang@etu.u-paris.fr, pascal.amsili@ens.fr
heather.susan.burnett@gmail.com

RÉSUMÉ

Nous présentons les résultats d'une expérience visant à savoir si la densité d'information (ou de surprise) affecte le redoublement du sujet dans des conversations spontanées. En utilisant la version française de GPT, nous estimons la surprise lexicale du sujet NP étant donné un contexte précédent et vérifions si la surprise du sujet affecte son redoublement. L'analyse de régression à effet mixte montre que, en plus des facteurs qui ont été montrés comme affectant le redoublement du sujet dans la littérature, la prévisibilité du sujet nominal est un prédicteur important du non-redoublement. Les sujets nominaux moins prédictibles tendent à être redoublés par rapport à ceux qui sont plus prédictibles. Notre travail confirme l'intérêt de l'hypothèse de l'Uniformité de la densité informationnelle (UID) pour le français et illustre l'opérationalisation de la densité informationnelle à l'aide de grands modèles neuronaux de langage.

ABSTRACT

The Uniform Information Density Hypothesis : the case of subjet-doubling in French

We present the results of an experiment investigating whether information density affects subject doubling in conversations in French. Using the French version of GPT, we estimate the lexical surprisal of the subject NP subject given a certain left context and verify whether the surprisal of the subject affects its doubling. A Mixed effect regression analysis shows that, in addition to factors that have been shown to affect subject doubling in the literature, the predictability of the NP is an important predictor of subject doubling. Less predictable NPs tend to be more often doubled with a clitic than more predictable ones. Our work thus provides additional support to the Uniform Information Density (UID) hypothesis in French and points to a way to the operationalization of information density with the help of large neural language models.

MOTS-CLÉS : uniformité de la densité informationnelle, redoublement du sujet, surprise, français oral, modèle Tranformer Génératif Pré-entraîné (GPT).

KEYWORDS: Uniform Information Density, subject doubling, surprisal, spoken French, Generative Pre-trained Transformer (GPT).

1 Introduction

Le redoublement du sujet à l'oral (1) est un phénomène fréquent en français, en particulier à l'oral, et il est conditionné par de nombreux paramètres linguistiques, qui vont de la phonologie à la structure

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 164–176.

informationnelle.

(1) Marie$_i$ elle$_i$ m'a prêté son vélo.

Ce phénomène se caractérise aussi par le fait qu'il n'apporte en général pas d'information nouvelle, les locuteurs ont donc le choix entre deux constructions sémantiquement comparables. Ce genre de phénomène d'alternance, que l'on peut aussi observer à propos de l'ordre des mots par exemple (2-a) *vs.* (2-b), est un terrain privilégié pour étudier l'influence d'un principe cognitif qui s'est révélé pertinent dans de nombreuses études (Maurits *et al.*, 2010; Cuskley *et al.*, 2021, parmi d'autres), le principe selon lequel les locuteurs choisissent la version qui uniformise la densité informationnelle de l'énoncé (hypothèse UID, définie et illustrée à la section 2).

(2) a. J'ai donné ce livre à Jade.
b. J'ai donné à Jade ce livre.

Le travail que nous présentons dans cet article s'inscrit dans cette lignée, et il vise à étudier la façon dont ce principe peut contribuer à expliquer les choix des locuteurs. À partir de corpus oraux récents annotés, nous proposons un modèle de régression logistique intégrant tous les facteurs connus pour avoir une influence sur le redoublement du sujet. Nous montrons que prendre en compte la dimension d'uniformité de la densité informationnelle augmente la capacité prédictive du modèle, ce qui conforte l'intérêt de cette hypothèse pour expliquer le phénomène.

Nous présentons à la section 2 l'hypothèse elle-même, et évoquons quelques travaux qui ont proposé de la prendre en compte. Le phénomène du redoublement du sujet fait l'objet de la section 3, où nous détaillons les variables dont l'influence a été établie, et proposons un premier modèle de régression logistique. Nous montrons à la section 4 qu'en formulant la densité informationnelle au moyen d'une probabilité obtenue par un modèle de langue, et en intégrant cette variable dans le modèle, on prédit mieux le redoublement du sujet. Les limites et perspectives ouvertes par cette étude sont présentées à la section 5.

2 L'hypothèse de l'uniformité de la densité informationnelle

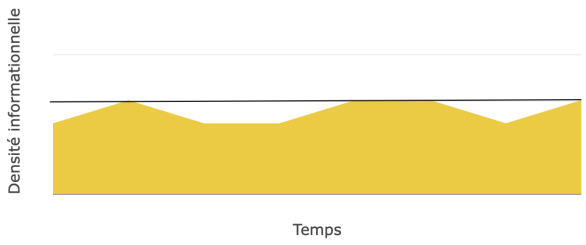


FIGURE 1 – Bonne utilisation du canal.

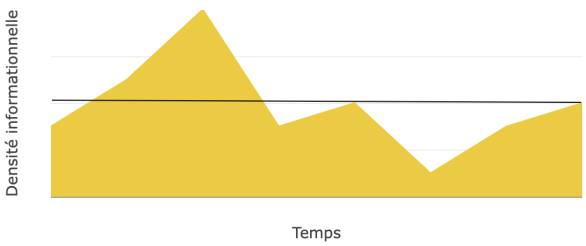


FIGURE 2 – Mauvaise utilisation du canal.

L'hypothèse de l'uniformité de la densité informationnelle (*Uniform Information Density*, UID) (Levy & Jaeger, 2007; Jaeger, 2010) pose que, dans la mesure où la grammaire le permet, les locuteurs distribuent l'information de la façon la plus homogène possible au sein d'une phrase. L'intuition

qui sous-tend cette hypothèse est double : d'une part, la communication peut être vue comme une transmission d'information à travers un canal bruité et limité par nos ressources cognitives, ce qui peut donner lieu à des erreurs, voire à l'échec de la transmission du message initial. Pour augmenter les chances de succès de la communication, il est préférable que l'information transmise à chaque unité de temps ne dépasse pas la limite de la capacité du canal (illustrée par une ligne noire dans les figures 1 et 2). D'autre part, une communication efficace suppose que le canal de communication ne soit pas sous-employé, et par conséquent qu'un niveau minimal d'informativité soit garanti. La figure 1 représente l'évolution de la densité informationnelle au cours du temps et illustre une communication proche de l'idéal selon UID, par constraste avec une utilisation sous-optimale illustrée à la figure 2.

On se place ici dans le cadre de la théorie de l'information (Shannon, 1948), qui permet de définir la densité informationnelle au moyen de la surprise pour chaque mot w_i, définie comme le logarithme négatif de la probabilité de l'apparition d'un mot étant donné le contexte (équation 1). Plusieurs études psycholinguistiques ont montré que les unités linguistiques présentant un niveau de surprise plus élevé sont associées à des difficultés de compréhension et nécessitent davantage de temps de lecture (Demberg & Keller, 2008; Wilcox *et al.*, 2020; Futrell *et al.*, 2020; Frank *et al.*, 2015).

$$I(w_i) = -\log P(w_i|w_0 \dots w_{i-1}) \tag{1}$$

Dans ce qui suit, nous parlerons de façon interchangeable de *surprise*, d'*informativité* ou de *niveau d'information* : par hypothèse, une unité peu prédictible (ayant donc une probabilité faible), est associée à un niveau élevé de surprise, et est donc considérée comme informative.

Diverses études portant sur des phénomènes de variation à différents niveaux linguistiques ont permis de conforter l'hypothèse UID. À titre d'illustration et de manière non exhaustive : niveau phonétique : un son ou un mot ayant une surprise plus élevée est prononcé plus lentement au sein et à travers les langues (Demberg *et al.*, 2012; Pimentel *et al.*, 2021) ; niveau syntaxique : les mots fonctionnels tendent à être omis lorsque la structure qu'ils introduisent est plus prédictible (par exemple, omission de *that* dans les complétives (Jaeger, 2010) et les relatives (Jaeger, 2011) en anglais, omission d'article dans les titres de journaux en allemand (Lemke *et al.*, 2017)) ; niveau discursif : l'omission des connecteurs est plus fréquente lorsque la relation discursive est moins surprenante (Torabi Asr & Demberg, 2015) ; entre autres.

Ces travaux montrent de façon convaincante l'importance de la densité informationnelle, mais la question de l'opérationalisation de l'hypothèse reste aujourd'hui encore délicate. Deux aspects sont particulièrement discutés dans la littérature. D'une part, il faut se demander si le calcul se fait localement ou globalement (voir (Meister *et al.*, 2021) pour une discussion approfondie). D'autre part, et de façon plus cruciale encore, il s'agit de savoir par quel moyen on va estimer le niveau d'informativité d'une unité (typiquement un token). Les études qui adoptent une approche de corpus utilisent souvent un modèle bigramme basé sur des fréquences (Jaeger, 2010, 2011; Lemke *et al.*, 2017). D'autres études se basent sur des indices plus linguistiquement motivés, mais plus spécifiques. Par exemple, Torabi Asr & Demberg (2015) s'appuient sur la présence d'une négation dans la première phrase en tant qu'indicateur approximatif de la prédictibilité d'une relation causale avec la deuxième phrase. Ces mesures prennent en considération un contexte assez limité et ne sont pas faciles à généraliser à d'autres phénomènes. Ainsi, dans l'article de Jaeger (2010) sur l'omission de *that*, la surprise d'une complétive est estimée par la fréquence de sous-catégorisation du verbe. Par exemple, *think* « penser » est souvent suivi d'une complétive et donc l'apparition d'une complétive est peu surprenante après ce verbe. Cependant, même pour un verbe donné, la probabilité de la

complétive peut varier selon les contextes. Par exemple, l'apparition d'une complétive après *think* est plus probable quand *I think* se trouve en position initiale de la phrase (3-a) que quand il se trouve en position non initiale (3-b).

(3) a. I think he is crazy.
b. He is crazy, I think.

Les modèles de langues sont largement utilisés dans des travaux de modélisation cognitive ou de psycholinguistique pour fournir des probabilités conditionnelles de mots étant donné le contexte gauche, afin de mesurer la corrélation entre la surprise des modèles et diverses mesures comportementales (temps de lecture, présence de difficultés de compréhension, mouvements occulaires, etc.). Des corrélations ont été observées dans de nombreuses études utilisant différents types de modèles tels que n-grammes, GRU (*Gated Recurrent Unit*), LSTM (*Long Short-Term Memory*) et plus récemment, GPT (*Generative Pre-trained Transformer*) (Goodkind & Bicknell, 2018; Wilcox *et al.*, 2020; Merkx & Frank, 2021; Kuribayashi *et al.*, 2022). Cependant, ils sont rarement utilisés pour explorer le rôle de la densité informationnelle dans la variation syntaxique. De plus, les phénomènes de variation restent peu étudiés en français sous cette optique (mais voir (Liang *et al.*, 2021) sur l'omission de *que* en français montréalais). Nous proposons d'utiliser un modèle de langue génératif, en l'occurrence GPT, pour estimer la probabilité (et donc l'information) des mots, afin de contribuer à cette ligne de recherche en français, sur un phénomène de redondance syntaxique, le redoublement du sujet.

3 Le phénomène

3.1 Redoublement du sujet

Le redoublement du sujet, où un sujet lexical et un clitique coréférentiel apparaissent en même temps dans une phrase ((4-a), à comparer à la version non-redoublée (4-b)), est un phénomène largement répandu en français oral [1].

(4) a. Mon père$_i$ il$_i$ est venu.
b. Mon père est venu.

De nombreux facteurs ont été proposés pour expliquer la variation entre les deux constructions (doublée *vs.* non-redoublée) : par exemple, le type de sujet nominal (Nadasdi, 1995; Auger, 1998; Auger & Villeneuve, 2010), le type de proposition (Auger & Villeneuve, 2010), la présence d'éléments intervenant entre le sujet et le verbe (Zahler, 2014), le statut informationnel (Zahler, 2014), entre autres. Malgré cette littérature riche sur le rôle des facteurs grammaticaux pertinents, peu d'études ont examiné ce phénomène sous l'angle de la théorie de l'information.

Pourtant, le redoublement peut être considéré comme un exemple de redondance syntaxique, un cas où l'hypothèse UID pourrait contribuer à expliquer le choix des locuteurs, dans la mesure où la présence du sujet clitique n'apporte pas d'information nouvelle à la phrase. On pourrait proposer le raisonnement suivant : le pronom redoublé est par essence peu informatif, et par conséquent introduire

1. Il est aussi possible de redoubler un sujet nominal postverbal par un sujet clitique (*Il$_i$ est venu mon père$_i$*). Dans la présente étude, nous nous limitons aux cas de redoublement de sujet préverbal.

un pronom conduit à une baisse locale du niveau de surprise. L'hypothèse UID prédit que l'on aura plus tendance à introduire ce pronom quand la surprise associée au sujet lexical est élevée, pour lisser le niveau de surprise, alors qu'au contraire avec un sujet lexical très prédictible, et donc de surprise peu élevée, l'ajout d'un pronom lui-même peu informatif n'a pas d'impact pertinent sur l'uniformité de la densité informationnelle.

Afin d'examiner la pertinence de cette proposition, nous avons mené une étude du redoublement du sujet dans un corpus de français oral, *Multicultural Parisian French* (MPF) (Gadet & Guerin, 2016; Gadet, 2017). Avant de procéder à l'opérationalisation de UID dans ce nouveau cas, il est important d'identifier et de contrôler les autres facteurs potentiels.

3.2 Une première modélisation

3.2.1 Extraction des données

Le corpus MPF se compose de 66 entretiens pour un total de près de 790 000 mots transcrits à ce jour, et vise à documenter le langage oral de jeunes âgés de 12 à 37 ans, issus d'un milieu familial multiculturel et résidant dans les banlieues parisiennes [2]. Il s'agit de conversations face à face entre amis ou connaissances portant sur des thèmes variés tels que la famille, la vie quotidienne, l'évolution des langues, entre autres. Le corpus n'étant pas muni d'annotations morpho-syntaxiques, nous l'avons d'abord segmenté et annoté en parties du discours à l'aide de *Stanza* (Qi *et al.*, 2020), puis utilisé l'*analyseur HOPS* (Grobol & Crabbé, 2021) pour en obtenir une analyse en dépendance. Nous avons ensuite procédé à l'extraction de toutes les phrases qui contiennent un sujet nominal (par exemple *mon père*, *un garçon*, *certains*, *tout le monde*, *Marie*...) de l'ensemble du corpus et indiqué si le sujet nominal est redoublé par un clitique (comme *il(s), elle(s), ce, ça*). Seuls les sujets préverbaux de troisième personne ont été pris en compte. Les pronoms forts, tels que *lui* et *eux*, ont été exclus. L'extraction a été faite à l'aide d'un script Python analysant les annotations morpho-syntaxiques et syntaxiques du corpus, complété par une vérification manuelle. Pour ce modèle comme pour le suivant, nous avons limité notre analyse aux cas où la tête du sujet est réduite à un seul token suite à la prétokenization par *Stanza* pour qu'il soit possible d'établir une valeur de surprise pour ce token. Ce processus a abouti à un jeu de données de 4 136 occurrences, dont le taux de redoublement est de 75,5 %.

3.2.2 Facteurs de contrôle

Un second script Python a été utilisé pour annoter de façon semi-automatique toutes les occurrences extraites pour les facteurs listés ci-dessous. L'objectif était de relever les facteurs qui conditionnent le redoublement du sujet, afin de les contrôler lors de l'exploration de l'effet de la densité informationnelle décrite dans la section suivante :

— Polarité de la phrase : affirmative, négative avec *ne*, négative sans *ne* ;
— Type de sujet : sujet quantifié universellement (par ex. *tout le monde*, *rien*), groupe nominal indéfini (par ex. *un garçon*), groupe nominal défini (par ex. *le garçon*, *Daniel*) ;
— Type de proposition : proposition principale, proposition subordonnée autre que les relatives, proposition relative ;

2. Les entretiens et les transcriptions sont en accès libre sur le site du corpus : https://www.ortolang.fr/market/corpora/mpf/v3 (mpf, 2019).

— Fréquence du verbe mesurée dans le même corpus ;
— Distance en nombre de mots entre la tête du sujet et le verbe.

3.2.3 Modélisation statistique

Une modélisation au moyen d'un modèle de régression logistique à effets mixtes a été réalisée avec le logiciel R Studio (Team, 2022), implémentée grâce à la fonction `glmer()` du package lme4 (Bates *et al.*, 2015). Les facteurs numériques sont transformés en logarithme et centrés. Pour les facteurs catégoriels, le codage de *Backward Difference* est employé, comparant les niveaux adjacents dans l'échelle définie ci-dessous, le niveau supérieur est comparé avec le précédent. En plus des effets fixes, le modèle prend en compte trois effets aléatoires : locuteur, lemme du verbe, lemme de la tête du sujet. Le modèle de référence est ainsi défini (redoublement = 1, non redoublement = 0) :

MODÈLE DE RÉFÉRENCE : Redoublement ~ polarité + type_sujet + type_proposition + fréquence_verbe + distance + (1 | lemma_verbe) + (1 | lemma_sujet) + (1 | locuteur)

La table 1 (première colonne, « modèle de référence ») récapitule les facteurs qui jouent un rôle significatif sur le redoublement du sujet dans le modèle de référence. Tous les facteurs de contrôle pris en compte sont significatifs.

	Modèle de référence				**Modèle principal**			
	Coef.	z	p	Sig.	Coef.	z	p	Sig.
(Intercept)	-2.96	-6.68	2.47e-11	***	-2.99	-6.76	1.35e-11	***
polarité (*ne* vs. aff.)	-6.43	-6.06	1.33e-09	***	-6.43	-6.06	1.36e-09	***
polarité (sans *ne* vs. *ne*)	6.87	6.39	1.62e-10	***	6.88	1.08	1.65e-10	***
sujet (ind. vs. uni.)	1.61	2.74	0.00608	**	1.62	2.80	0.00517	**
sujet (defini vs. ind.)	2.47	5.28	1.30e-07	***	2.44	5.28	1.26e-07	***
prop. (sub. vs. prin.)	1.43	4.42	9.95e-06	***	1.45	4.48	7.37e-06	***
prop. (rel. vs. sub.)	0.97	7.86	3.78e-15	***	0.95	7.64	2.19e-14	***
fréquence verbe	0.27	3.18	0.00148	**	0.26	3.10	0.00193	**
distance tête suj. - verbe	0.26	4.46	8.21e-06	***	0.26	4.51	6.53e-06	***
surprise (cf. § 4.2)	-	-	-	-	**0.15**	**2.23**	**0.02566**	*

TABLE 1 – Modèle de régression logistique à effets mixtes du redoublement du sujet en fonction des facteurs de contrôle (modèle de référence) et de la surprise du sujet (modèle principal), avec les coefficients de pente, les valeurs *z*, les valeurs *p* ainsi que les niveaux de significativité des effets fixes. Le locuteur, les lemmes du sujet et du verbe ont été considérés comme des intercepts aléatoires.

Les effets des différents facteurs de contrôle dans le modèle de référence (table 1) peuvent être résumés brièvement :

— Polarité de la phrase : les phrases négatives contenant « ne » défavorisent le redoublement du sujet en comparaison avec les phrases négatives sans « ne » et les phrases affirmatives.
— Type de sujet : les sujets quantifiés universellement sont redoublés le moins souvent, tandis que les sujets définis sont redoublés le plus souvent. Les sujets indéfinis se situent entre les deux.
— Type de proposition : les propositions principales sont corrélées avec un taux d'omission le plus élevé, tandis que les subordonnées défavorisent le redoublement. Parmi les subordonnées,

les propositions relatives sont celles qui défavorisent le plus le redoublement et les autres types de subordonnées se situent entre les deux.
— Fréquence du verbe : plus un verbe est fréquent dans le corpus, plus il y a de chances que le sujet soit redoublé.
— Distance en nombre de mots entre la tête du sujet et le verbe : plus le verbe est distant de la tête du sujet, plus il est probable que le sujet soit redoublé.

4 Prise en compte de la densité informationnelle

4.1 Utilisation d'un modèle de langue génératif

Afin d'estimer la surprise du sujet nominal, nous avons utilisé un modèle GPT (transformer génératif pré-entraîné). Basés sur les transformers (Vaswani *et al.*, 2017), et utilisant des blocs de type décodeur (i.e. avec auto-attention masquée), les modèles GPT sont capables de générer un token à la position t_i à partir d'un préfixe $t_0 ... t_{i-1}$ qui peut être très long (1 024 tokens au maximum). Puisque le modèle GPT est capable de prendre en considération un contexte plus étendu en comparaison avec d'autres types de modèles tels que n-gramme et LSTM, nous l'avons choisi dans la présente étude. Toutefois, il a été constaté que les valeurs de surprise estimées par un modèle GPT ayant plus de paramètres sont moins efficaces pour prédire les temps de lecture de mots, en comparaison avec ses versions plus petites (Oh *et al.*, 2022; Oh & Schuler, 2023). De ce fait, notre étude est basée sur le modèle GPT le plus petit disponible pour le français, avec 124 millions de paramètres (Simoulin & Crabbé, 2021), GPT$_{fr}$-124M, lui-même adapté des modèles OpenAI GPT et GPT-2 (Radford *et al.*, 2019a,b).

4.1.1 Ajustement du modèle

Comme GPT$_{fr}$-124M a été entraîné principalement sur les documents écrits, il est moins capable de prendre en considération les caractéristiques des textes oraux (hésitations, répétitions, reformulations, registre lexical familier...). Par conséquent, nous avons procédé à un ajustement du modèle (*fine-tuning*) avec un autre corpus du français parisien parlé, le Corpus de Français Parlé Parisien des années 2000 (CFPP2000) (Branca-Rosoff *et al.*, 2012). Collecté à partir de 2005-2006, il se compose d'un ensemble d'entretiens réalisés dans différents quartiers de Paris et de la proche banlieue. Actuellement, 51 entretiens transcrits contenant un total de 750 000 tokens (d'après la prétokenization par *Stanza*) sont disponibles sur le site [3], à partir desquels le corpus d'entraînement et d'évaluation a été construit pour l'ajustement du modèle. La tâche est de prédire le prochain mot vu le contexte précédent. Afin de ne pas dégrader la généralité du modèle, pendant l'ajustement, nous avons utilisé la même tokenization définie par le modèle GPT$_{fr}$-124M (Simoulin & Crabbé, 2021) et construit le corpus d'entraînement d'une façon similaire.

Chaque entretien est divisé en tours de parole. Un exemple d'apprentissage est construit à partir de tours de parole successifs concaténés dans la limite de 1 024 tokens. Une fois le nombre atteint, un nouvel exemple est construit à partir du tour de parole suivant, et aucun exemple ne franchit la frontière des entretiens (le dernier exemple est complété (*padding*)). Nous avons divisé les entretiens en corpus d'entraînement et de test, contenant respectivement 583 (46 entretiens) et 97 exemples (5 entretiens). L'ajustement s'est déroulé pendant 30 époques avec les paramètres par défaut.

3. http://cfpp2000.univ-paris3.fr/

4.1.2 Évaluation du modèle ajusté

Le modèle ajusté est évalué sur le corpus de test du CFPP2000 (10% des entretiens) et sur 50% des entretiens de MPF respectivement. Nous avons évalué la perplexité sur la base de la tokenization qui est identique pour le modèle original (GPT$_{fr}$-124M) et le modèle ajusté. Comme on le voit dans la table 2, l'ajustement a permis une réduction significative dans la perplexité évaluée sur les exemples de test dans les deux corpus, suggérant que le modèle ajusté s'adapte mieux aux données orales que le modèle original.

	CFPP2000	MPF
nombre d'exemples de test	97	379
perplexité moyenne du modèle original	42,25	42,61
perplexité moyenne du modèle ajusté	28,93	40,97
p valeur	$< 2.2e-16$	0,043

TABLE 2 – Comparaison de la perplexité par les deux modèles dans les deux corpus oraux. Des t-tests ont été employés pour tester la significativité de la différence.

4.1.3 Estimation de l'information du sujet

Grâce à ce modèle ajusté, il est possible d'estimer la probabilité (et donc la surprise) du sujet nominal dans toutes les occurrences extraites du corpus. Afin de faciliter le calcul, on calcule la probabilité seulement pour la tête du syntagme nominal (dont on s'est assuré qu'elle était mono-lexicale, voir plus haut). Il reste à établir la longueur du contexte gauche que nous devons fournir au modèle de langue pour produire cette probabilité. Comme le sujet d'une phrase est souvent situé en position initiale d'une phrase, ce qui rend l'estimation de la probabilité peu fiable, il convient d'élargir le contexte considéré. Généralement, ce genre de modèle génératif est d'autant plus performant que le contexte gauche est étendu. Cependant, ce que nous cherchons à savoir est plutôt quelle taille du contexte à gauche serait la plus appropriée pour simuler le comportement humain. Dans une étude récente, Kuribayashi *et al.* (2022) montrent que plus le contexte précédent est limité, plus la probabilité du mot estimée par le modèle GPT est corrélée avec le temps de lecture moyen de ce mot mesuré par oculométrie en anglais et en japonais. C'est la raison pour laquelle, dans la présente étude, nous avons pris comme contexte gauche un seul tour de parole avant la cible, indépendamment du nombre de mots (voir une illustration à la table 3). Nous revenons en conclusion sur la question de la taille du contexte gauche. Un tour de parole compte en moyenne 10 tokens. La tâche consiste à prédire le mot à la position de la tête du sujet en position t_i (en gras). La probabilité logarithmique négative du sujet sera considérée comme la surprise du sujet.

Contexte ($t_0 ... t_{i-1}$)		Tête du sujet (t_i)	$P(t_i\|\text{contexte})$	$-\log P(t_i\|\text{contexte})$
Tour précédant la cible	*Cible*			
Oui c'est mes origines.	Ben c'est moi quoi chez moi euh on ma	**mère**	0,0125	4,3782

TABLE 3 – Exemple de l'estimation de la surprise du sujet en prenant en compte un tour de parole.

Puisque le modèle GPT utilise un vocabulaire de type *bytepair encoding* (BPE), il est possible que la tête du sujet soit décomposée en sous-mots. Dans la présente étude, 21 % des tokens du sujet ont été décomposés en 2 à 6 sous-mots. Il s'agit souvent de noms propres, de mots contenant un symbole

(comme *quelqu'un*, *grand-mère*) ou de mots familiers (verlan et emprunts). Puisque la fréquence des sous-mots est plus élevée que celle du mot en entier, la moyenne des probabilités sous-estimerait la surprise de l'ensemble du mot. Par conséquent, nous définissons la probabilité d'un mot comme la probabilité conjointe des sous-mots qui le composent. La surprise de la tête du sujet est donc définie comme la somme des log probabilités de ses sous-mots, ce qui est conforme aux pratiques de plusieurs études analysant la plausibilité cognitive des modèles de langue neuronaux (Wilcox *et al.*, 2020; Kuribayashi *et al.*, 2021; Oh & Schuler, 2022).

Nous utilisons le même modèle de régression logistique que celui de la section précédente, auquel est ajoutée la variable correspondant à la surprise de la tête du sujet telle qu'elle est estimée par le modèle GPT.

4.2 Résultats

Les résultats de ce nouveau modèle de régression logistique sont présentés en colonne de droite de la table 1 (déjà donnée à la fin de la section 3). Ils montrent que, en plus des effets identifiés précédemment, la surprise de la tête du sujet a un effet sur le taux de redoublement. Pour étudier plus finement la relation entre surprise et probabilité de redoublement, nous représentons à la figure 3 toutes les observations regroupées par valeur de surprise en groupes de 30 contre la proportion de redoublement. On voit que moins un sujet est prédictible (plus il est informatif, donc), plus il a tendance à être redoublé par un clitique. Un test anova confirme que l'inclusion de la surprise de la tête du sujet dans le modèle contribue à une amélioration de performance significative ($p < 0.05$).

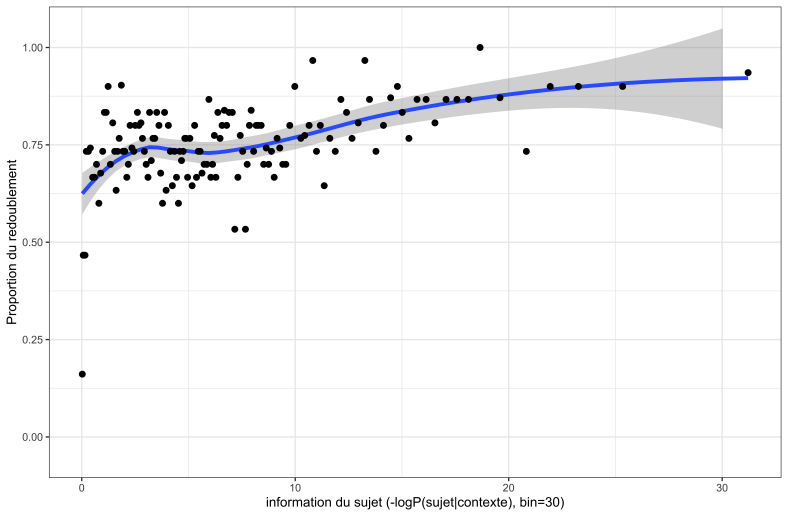


FIGURE 3 – Taux de redoublement du sujet en fonction de la surprise de la tête du sujet nominal. Chaque point représente un groupe de 30 observations regroupées par niveau de surprise. La courbe est générée par la méthode *loess*.

5 Conclusion

5.1 Nouveau champ d'aplication de UID

Le résultat montre une relation entre le taux de redoublement et la surprise du sujet nominal, ce qui suggère que l'hypothèse UID contribue à expliquer le phénomène du redoublement du sujet étudié ici. On peut en effet supposer que, comme la densité informationnelle au niveau du sujet risque de dépasser la capacité du canal quand le nom est moins prédictible, les locuteurs préfèrent utiliser un clitique qui n'apporte rien sur l'information du sujet, afin de réduire la densité informationnelle. En revanche, lorsque le nom en position de sujet est très prédictible, l'ajout d'un clitique conduirait à une densité plus basse, et explique donc que la version sans clitique soit préférable. On peut noter cependant que le taux de redoublement est très élevé en général (75,5 %), ce qui suggère que le sujet clitique pourrait être en voie d'être grammaticalisé comme un marqueur d'accord faisant partie de la flexion du verbe en français informel (Auger, 1995; Culbertson, 2010).

5.2 Limites et perspectives

Bien que nos résultats confirment l'intérêt de l'UID pour le français, il convient de remarquer que l'effet de la densité informationnelle du sujet est faible, son coefficient étant le plus petit parmi les facteurs testés (table 1). De plus, cet effet est sensible aux effets aléatoires. En effet, lorsque le lemme du sujet est exclu de l'analyse, la corrélation entre la surprise et le redoublement devient plus forte dans le modèle mixte (coefficient : 0.18, $p < 0.001$). Il est probable que cela tienne en partie au fait que dans nos données, 60,6 % des noms (lemmatisés) n'apparaissent qu'une fois comme sujet et 91,7 % des noms (lemmatisés) apparaissent moins de 5 fois comme sujet (au total 1 178 différents lemmes du sujet). Pour ces cas-là, il n'est pas surprenant que l'intercept aléatoire du lemme du sujet explique mieux la variance du sujet en ce qui concerne le redoublement et diminue donc la force d'explication de l'effet de la surprise du sujet en général.

Par ailleurs, il s'avère que les résultats sont sensibles à la taille de fenêtre du contexte considéré. En effet, la performance du modèle GPT en termes de prédiction de sujet s'améliore si on accroit la taille du contexte, jusqu'à une exactitude maximale de 28,7% avec le contexte le plus large testé. Toutefois, il a aussi été observé que l'effet de la surprise du sujet diminue progressivement à mesure que le modèle GPT a accès à un contexte plus large lors de l'estimation de la probabilité du sujet. Cela pourrait s'expliquer par le fait que plus le contexte est large, plus le modèle est certain du choix lexical du sujet, ce qui rend la distribution de surprises moins équilibrée. Par ailleurs, Kuribayashi *et al.* (2022) mettent en évidence l'existence d'un écart entre la capacité d'accès au contexte des modèles et des humains, et montrent qu'une limitation du contexte rend le comportement des modèles de langue plus similaires à ceux des humains. Dans une autre étude, Kuribayashi *et al.* (2021) montrent aussi qu'un modèle de langue avec une perplexité plus basse (donc plus performant) reflète mieux le comportement humain pour l'anglais, mais pas pour le japonais. Des investigations supplémentaires sont nécessaires afin d'explorer l'impact de la taille du contexte sur le temps de lecture en français, en vue d'identifier la taille de fenêtre la plus appropriée pour l'estimation de la surprise.

Notre travail met donc en évidence l'intérêt que peut présenter l'utilisation d'un modèle de langue pour estimer le niveau de surprise des mots dans une phrase. Nous pensons que ce type de modèle présente l'intérêt d'être très général, d'incorporer des connaissances linguistiques grâce au pré-apprentissage, et de permettre la prise en compte d'un contexte gauche assez large, et devrait donc être privilégié par

rapport à différentes opérationalisations de l'hypothèse UID qui ont été proposées et qui n'ont pas ces qualités. [4]

Remerciements

Nous remercions Marie Candito pour sa relecture d'une version précédente de ce papier et ses conseils. Nous tenons également à remercier Vera Demberg pour une discussion sur l'analyse statistique, ainsi que Benoît Crabbé pour son aide dans l'analyse en relations de dépendance du corpus étudié. Ces travaux ont bénéficié du financement de l'ERC dans le cadre du programme de recherche et d'innovation Horizon 2020 de l'Union européenne (N°850539).

Références

(2019). MPF. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

AUGER J. (1995). Les clitiques pronominaux en français parlé informel : une approche morphologique. *Revue québécoise de linguistique*, **24**(1), 21–60. DOI : 10.7202/603102ar.

AUGER J. (1998). Le redoublement des sujets en français informel québécois : Une approche variationniste. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, **43**(1), 37–63. DOI : 10.1017/S0008413100020429.

AUGER J. & VILLENEUVE A.-J. (2010). La double expression des sujets en français saguenéen : Étude variationniste. *Hétérogénéité et Homogénéité dans les pratiques langagieres : Mélanges offerts à Denise Deshaies. Quebec : Presses de l'Université Laval*, p. 67–86.

BATES D., MÄCHLER M., BOLKER B. & WALKER S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, **67**(1), 1–48. DOI : 10.18637/jss.v067.i01.

BRANCA-ROSOFF S., FLEURY S., LEFEUVRE F. & PIRES M. (2012). Discours sur la ville. Présentation du Corpus de Français Parlé Parisien des années 2000 (CFPP2000). *article en ligne,* `http://cfpp2000.univ-paris3.fr/Articles.html`.

CULBERTSON J. (2010). Convergent evidence for categorial change in French : From subject clitic to agreement marker. *Language*, **86**(1), 85–132. DOI : 10.1353/lan.0.0183.

CUSKLEY C., BAILES R. & WALLENBERG J. (2021). Noise resistance in communication : Quantifying uniformity and optimality. *Cognition*, **214**, 104754. DOI : 10.1016/j.cognition.2021.104754.

DEMBERG V. & KELLER F. (2008). Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, **109**(2), 193–210. DOI : 10.1016/j.cognition.2008.07.008.

DEMBERG V., SAYEED A. B., GORINSKI P. J. & ENGONOPOULOS N. (2012). Syntactic surprisal affects spoken word duration in conversational contexts. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, p. 356–367, Jeju Island, Korea.

4. Nous avons cependant testé, en manière de contrôle, une mesure beaucoup plus grossière pour estimer la surprise du sujet : la fréquence de la tête du sujet (lemmatisé) dans le corpus MPF. En remplaçant la surprise calculée par GPT par cette mesure dans le modèle, nous avons observé une corrélation négative entre la fréquence du sujet et le redoublement, ce qui est de nouveau compatible avec l'UID, car un sujet plus fréquent est plus prédictible et donc moins informatif en général. Ce résultat n'est pas surprenant puisque, comme nous l'avons vérifié, la fréquence du lemme et la probabilité donnée par GPT sont corrélés (test de Pearson : cor : -0.58, $p < 2.2e-16$).

FRANK S. L., OTTEN L. J., GALLI G. & VIGLIOCCO G. (2015). The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, **140**, 1–11. DOI : 10.1016/j.bandl.2014.10.006.

FUTRELL R., GIBSON E. & LEVY R. P. (2020). Lossy-Context Surprisal : An Information-Theoretic Model of Memory Effects in Sentence Processing. *Cognitive Science*, **44**(3). DOI : 10.1111/cogs.12814.

GADET F., Éd. (2017). *Les Parlers Jeunes Dans l'île-de-France Multiculturelle*. Paris and Gap : Ophrys.

GADET F. & GUERIN E. (2016). Construire un corpus pour des façons de parler non standard : « Multicultural Paris French ». *Corpus*, **15**. DOI : 10.4000/corpus.3049.

GOODKIND A. & BICKNELL K. (2018). Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, p. 10–18, Salt Lake City, Utah : Association for Computational Linguistics. DOI : 10.18653/v1/W18-0102.

GROBOL L. & CRABBÉ B. (2021). Analyse en dépendances du français avec des plongements contextualisés (French dependency parsing with contextualized embeddings). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 106–114, Lille, France : ATALA.

JAEGER T. F. (2010). Redundancy and reduction : Speakers manage syntactic information density. *Cognitive Psychology*, p. 23–62.

JAEGER T. F. (2011). Corpus-based research on language production : Information density and reducible subject relatives. *Language from a cognitive perspective : grammar, usage and processing. Studies in honor of Tom Wasow*, p. 161–198.

KURIBAYASHI T., OSEKI Y., BRASSARD A. & INUI K. (2022). Context Limitations Make Neural Language Models More Human-Like. In *EMNLP* : arXiv. http://arxiv.org/abs/2205.11463.

KURIBAYASHI T., OSEKI Y., ITO T., YOSHIDA R., ASAHARA M. & INUI K. (2021). Lower Perplexity is Not Always Human-Like. In *ACL* : arXiv. http://arxiv.org/abs/2106.01229.

LEMKE R., HORCH E. & REICH I. (2017). Optimal encoding ! - Information Theory constrains article omission in newspaper headlines. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, p. 131–135, Valencia, Spain : Association for Computational Linguistics.

LEVY R. & JAEGER T. (2007). Speakers optimize information density through syntactic reduction. In B. SCHÖLKOPF, J. PLATT & T. HOFMANN, Éds., *Advances in Neural Information Processing Systems 19 : Proceedings of the 2006 Conference*, volume 19, p. 849–856 : MIT Press.

LIANG Y., AMSILI P. & BURNETT H. (2021). New ways of analyzing complementizer drop in Montréal French : Exploration of cognitive factors. *Language Variation and Change*, **33**(3), 359–385. DOI : 10.1017/S0954394521000223.

MAURITS L., NAVARRO D. & PERFORS A. (2010). Why are some word orders more common than others ? A uniform information density account. In *Advances in Neural Information Processing Systems*, volume 23 : Curran Associates, Inc.

MEISTER C., PIMENTEL T., HALLER P., JÄGER L., COTTERELL R. & LEVY R. (2021). Revisiting the Uniform Information Density Hypothesis. *arXiv :2109.11635 [cs]*.

MERKX D. & FRANK S. L. (2021). Human Sentence Processing : Recurrence or Attention ? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, p. 12–22, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.cmcl-1.2.

NADASDI T. (1995). Subject NP doubling, matching, and minority French. *Language Variation and Change*, **7**(1), 1–14. DOI : 10.1017/S0954394500000879.

OH B.-D., CLARK C. & SCHULER W. (2022). Comparison of Structural Parsers and Neural Language Models as Surprisal Estimators. *Frontiers in Artificial Intelligence*, **5**.

OH B.-D. & SCHULER W. (2022). Entropy- and Distance-Based Predictors From GPT-2 Attention Patterns Predict Reading Times Over and Above GPT-2 Surprisal. In *EMNLP* : arXiv. http://arxiv.org/abs/2212.11185.

OH B.-D. & SCHULER W. (2023). Why Does Surprisal From Larger Transformer-Based Language Models Provide a Poorer Fit to Human Reading Times ? *Transactions of the Association for Computational Linguistics*, **11**, 336–350. DOI : 10.1162/tacl_a_00548.

PIMENTEL T., MEISTER C., SALESKY E., TEUFEL S., BLASI D. & COTTERELL R. (2021). A surprisal–duration trade-off across and within the world's languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 949–962, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.73.

QI P., ZHANG Y., ZHANG Y., BOLTON J. & MANNING C. D. (2020). Stanza : A Python Natural Language Processing Toolkit for Many Human Languages. *arXiv :2003.07082 [cs]*.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2019a). Improving Language Understanding by Generative Pre-Training.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019b). Language Models are Unsupervised Multitask Learners. Open AI Technical Report.

SHANNON C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, **27**(3), 379–423.

SIMOULIN A. & CRABBÉ B. (2021). Un modèle Transformer Génératif Pré-entrainé pour le_____ français (Generative Pre-trained Transformer in_____ (French). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 246–255, Lille, France : ATALA.

TEAM R. C. (2022). R : A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.

TORABI ASR F. & DEMBERG V. (2015). Uniform Surprisal at the Level of Discourse Relations : Negation Markers and Discourse Connective Omission. In *Proceedings of the 11th International Conference on Computational Semantics*, p. 118–128, London, UK : Association for Computational Linguistics.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention Is All You Need. In *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA.

WILCOX E. G., GAUTHIER J., HU J., QIAN P. & LEVY R. (2020). On the predictive power of neural language models for human real-time comprehension behavior. In *CogSci*. https://arxiv.org/abs/2006.01912.

ZAHLER S. (2014). Variable subject doubling in spoken Parisian French. *University of Pennsylvania Working Papers in Linguistics*, **20**(1), 38.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Augmentation des modèles de langue français par graphes de connaissances pour la reconnaissance des entités biomédicales

Aidan Mannion[1,2] Didier Schwab[1] Lorraine Goeuriot[1] Thierry Chevalier[3]
(1) Laboratoire d'Informatique de Grenoble, Univ. Grenoble Alpes, CNRS, 38058 Grenoble, France
(2) EPOS SAS, 2-4 Boulevard Des Îles, 92130 Issy-les-Moulineaux, France
(3) UFR de Médecine Univ. Grenoble Alpes, Domaine de la Merci, 38700 La Tronche, France
prénom.nom@univ-grenoble-alpes.fr

RÉSUMÉ

Des travaux récents dans le domaine du traitement du langage naturel ont démontré l'efficacité des modèles de langage pré-entraînés pour une grande variété d'applications générales. Les modèles de langage à grande échelle acquièrent généralement ces capacités en modélisant la distribution statistique des mots par un apprentissage auto-supervisé sur de grandes quantités de texte. Toutefois, pour les domaines spécialisés à faibles ressources, tels que le traitement de documents cliniques, la nécessité d'intégrer des connaissances structurées reste d'une grande importance. Cette nécessité est d'autant plus grande pour les langues autres que l'anglais. Cet article se concentre sur l'une de ces applications spécialisées de la modélisation du langage à partir de ressources limitées : l'extraction d'informations à partir de documents biomédicaux et cliniques en français. En particulier, nous montrons qu'en complétant le pré-entraînement en mots masqués des réseaux neuronaux transformer par des objectifs de prédiction extraits d'une base de connaissances biomédicales, leurs performances sur deux tâches différentes de reconnaissance d'entités nommées en français peuvent être augmentées.

ABSTRACT

Unsupervised language model augmentation via knowledge graphs for French biomedical entity recognition

Recent work in natural language processing has demonstrated the effectiveness of large pre-trained language models for a wide variety of general applications. Large language models typically acquire these capabilities by modelling the statistical distribution of words via self-supervised training on large amounts of text. However, for low-resource settings, such as clinical document processing, particularly in languages other than English, the need to integrate structured domain knowledge remains of high importance. This paper focuses on one such low-resource specialised application of language modelling - information extraction from French-language clinical & biomedical documents. In particular, we show that supplementing the masked-language pre-training of transformer neural networks with prediction objectives extracted from structured biomedical knowledge graphs improves the downstream performance on two different named entity recognition tasks in French.

MOTS-CLÉS : TALN biomédical/clinique, extraction des informations, apprentissage automatique supervisé et auto-supervisé.

KEYWORDS: Biomedical/clinical NLP, information extraction, supervised & self-supervised machine learning.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 177–189.

1 Introduction

Les développements des dernières années en informatique et science des données représentent des pistes de recherche assez prometteuses pour le domaine de la santé, et l'utilisation généralisée des dossiers de santé électroniques (DSÉ) dans l'informatique médicale a beaucoup augmenté la disponibilité de dossiers cliniques en texte libre qui peuvent potentiellement être utilisés pour aider à la prise de décision par les professionel.les de santé (Wang *et al.*, 2019). Simultanément, les progrès dans le traitement automatique du langage naturel (TALN), notamment des approches basées sur l'apprentissage profond, ont rendu possible un traitement du texte beaucoup plus sophistiqué et efficace, et ont facilité l'amélioration de l'état de l'art dans plusieurs domaines d'application. En particulier, la modélisation de langage générale par des réseaux de neurones *transformer* (Vaswani *et al.*, 2017) s'est avérée très efficace pour de nombreuses applications. L'utilité de cette modélisation repose sur le paradigme d'apprentissage par transfert, où les représentations des données apprises sur une certaine tâche d'apprentissage, habituellement auto-supervisée, peuvent être utilisées d'une manière performante pour une autre tâche dans le même domaine d'application. Cette approche est d'autant plus utile pour les domaines à faibles ressources, tels que le domaine du traitement des textes cliniques, pour lesquels la quantité de données disponibles pour entraîner des algorithmes sur des tâches spécifiques n'est souvent pas suffisante (Dubois *et al.*, 2017). La communauté de la recherche biomédicale fait alors un usage de plus en plus intensif de ces techniques d'exploration de texte pour toute une série d'applications différentes (Ayala Solares *et al.*, 2020). Toutefois, la modélisation de langage faite uniquement à partir du texte libre se trouve souvent insuffisante pour bien intégrer dans ces modèles les connaissances du domaine clinique qui sont nécessaires pour un traitement efficace et fiable des documents cliniques (Sushil *et al.*, 2021). Dans ce travail, nous nous concentrons sur la tâche de la reconnaissance des entités nommées, l'une des tâches supervisées les plus étudiées dans le domaine du traitement des textes cliniques. Le travail d'adaptation des modèles de langage est divisé en trois parties :

— L'adaptation des modèles pour le domaine clinique par pré-entraînement sur un corpus de cas cliniques (section 2.2),
— L'intégration des connaissances du domaine par l'introduction des tâches supplémentaires - classification des triplets et prédiction des liens - basées sur le graphe de connaissances UMLS (Bodenreider (2004), section 2.3).
— L'adaptation des modèles pour les tâches de classification des mots par rapport à des catégories spécifiques au domaine (sections 2.4 et 3).

Nous étudions les performances de trois variantes de l'encodeur de texte BERT (Devlin *et al.*, 2019) sur trois tâches de reconnaissance des entités nommées. Les trois variantes pour lesquelles nous présentons des résultats dans ce papier sont CamemBERT (Martin *et al.*, 2020), FlauBERT (Le *et al.*, 2020), et une version de l'encodeur entraînée *from-scratch* sur un corpus de cas cliniques en français. Une description plus détaillée des corpus utilisés se trouve dans la section 1.3. Nous mettons à disposition le code et les modèles utilisés lors de nos expériences.

1.1 Encodeurs transformer pour le domaine clinique

Il existe plusieurs études qui adaptent l'architecture neuronale des transformers pour le domaine médical ; notamment Alrowili & Shanker (2021); Li *et al.* (2020); Lee *et al.* (2020) et Alsentzer *et al.* (2019). Ces travaux ont tous introduit des nouvelles variantes du modèle BERT et montrent des résultats au niveau de l'état de l'art sur des tâches telles que la prédiction de mortalité des patients, la

reconnaissance des entités médicales, la classification des documents, etc. Les travaux listés ci-dessus sont tous basés sur la langue anglaise et assez peu de projets ont été réalisés pour le français. Bien que le développement d'un modèle de langage français spécialisé pour le domaine biomédical soit un sujet de recherche très pertinent sur lequel quelques travaux ont déjà été menés (Berhe *et al.*, 2022; Dura *et al.*, 2022; El Boukkouri *et al.*, 2020), il n'existe pas de modèle de langage clinique en français librement disponible à la communauté de recherche, au moment de la rédaction de cet article. Il existe aussi quelques études sur l'adaptation des transformers français à des tâches de traitement de textes médicaux comme la reconnaissance des entités (Le Clercq de Lannoy *et al.*, 2022; Copara *et al.*, 2020) et la classification des documents (Chenais *et al.*, 2021), mais il s'agit d'un domaine qui n'a pas encore été pleinement exploré.

1.2 Travaux connexes : intégration des bases de connaissances biomédicales

Une grande partie des données liées au domaine de la santé et de la médecine est stockée sous une forme structurée, comme les ontologies ou les graphes de connaissances. L'importance d'exploiter ces données pour des applications d'apprentissage machine est donc largement reconnu dans ces domaines, particulièrement pour le traitement du texte (Chang *et al.*, 2020; Nicholson & Greene, 2020). Il a été démontré que les méthodes d'apprentissage combinant des représentations neuronales de texte avec des données biomédicales structurées améliorent les résultats obtenus dans une large gamme de tâches du traitement du texte médical en anglais (Naseem *et al.*, 2022; Meng *et al.*, 2021; Roy & Pan, 2021). Plusieurs stratégies pour intégrer les graphes de connaissances dans les modèles de langue de type BERT existent. Ces méthodes peuvent être divisées en deux types d'approches : **1)** des modifications de l'architecture neuronale du modèle pour intégrer des plongements de graphes de connaissance dans le même espace vectoriel que les plongements de mots (Peters *et al.* (2019), par exemple), et **2)** des modifications des données et de l'objectif de pré-entraînement pour que le modèle prenne en compte les informations contenues dans la base de connaissances en construisant ses plongements de mots, sans changer la structure interne du transformer. Dans ce travail, nous nous limitons aux approches du deuxième type, en nous positionnant dans un paradigme centré sur les données (Hamid, 2022). Autrement dit, nous exploitons uniquement la flexibilité des transformers de s'adapter à plusieurs modalités de données et plusieurs objectifs d'apprentissage. Nous pouvons conclure qu'avec des architectures qui sont généralisables de telle manière pour la modélisation de langage, il n'y a pas toujours besoin de changer leur structure pour les adapter aux types de données autres que le texte.

1.3 Données utilisées

Corpus de texte clinique Pour le pré-entraînement par mots masqués des modèles, nous utilisons la partie française du corpus E3C (European Clinical Case Corpus, Minard *et al.* (2021)), mis à disposition par l'organisation ELG (European Language Grid) [1]. Ce corpus consiste en des descriptions des cas cliniques tirés de revues médicales en français ainsi que des documents d'information sur les médicaments en provenance d'une base de données publique française. Nous nous limitons à ce corpus uniquement pour montrer l'efficacité de ces approches, même dans les situations pour lesquelles une quantité relativement faible de textes est disponible. Au total, il contient 25 740 documents. Pour nettoyer les documents de ce corpus, nous avons simplement enlevé les URLs et supprimé les phrases

1. https://live.european-language-grid.eu/catalogue/corpus/7618

contenant moins de quatre mots. Après ce nettoyage, le corpus entier de pré-entraînement se compose de 63,7 millions de mots et 2,7 millions de phrases.

Base de connaissances médicales Nous exploitons la base de connaissances UMLS (Système de langage médical unifié, (Bodenreider, 2004)) pour l'entraînement supplémentaire sur les données structurées. La partie française du métathésaurus UMLS (version 2022AB) contient 203 059 noms de concepts, catégorisés par 60 764 identifiants uniques (CUIs) et liés les uns aux autres par presque 1,5 millions de relations sémantiques structurées.

Corpus d'évaluation Nous utilisons deux tâches de classification des mots pour évaluer la capacité des modèles de reconnaître les entités médicales dans les documents cliniques, à partir des corpus médicaux annotés suivants :

1. **QUAERO :** (Névéol *et al.*, 2014) Pour cette évaluation, nous disposons du corpus de titres MEDLINE annotés dans le cadre de QUAERO, une ressource pour la reconnaissance et la normalisation des entités médicales. Ces titres sont annotés au niveau des mots avec des CUIs et des regroupements sémantiques d'UMLS.
2. **CAS-DEFT :** Ce corpus consiste en des annotations du corpus CAS (Grabar *et al.*, 2018) qui ont été mises à disposition dans le contexte du Défi Fouilles de Texte (DEFT) 2021 (Grouin *et al.*, 2021). Nous utilisons les annotations des mots en fonction des groupes sémantiques UMLS comme les étiquettes d'entraînement pour cette tâche.

Les tailles des partitions de ces corpus utilisées dans nos expériences sont détaillées dans le tableau 4.

TABLE 1 – Nombre de documents utilisés pour l'affinage des modèles de langue pour la reconnaissance des entités médicales.

	train	dev	test
QUAERO	788	790	787
CAS-NER	167	54	54

2 Méthodologie

2.1 Le Métathésaurus UMLS

Le métathésaurus UMLS est constitué de grandes quantités de concepts biomédicaux recoltés à partir de nombreuses sources d'informations biomédicales ontologiques. L'un de ses aspects les plus utiles est la manière dont il fournit des liens sémantiques entre des paires de concepts provenant de systèmes terminologiques différents. Cela permet notamment de regrouper sous le même identifiant de concept plusieurs termes différents provenant de vocabulaires médicaux différentes. Nous considérons le métathésaurus UMLS comme un graphe $\mathcal{G} = (C, R, E)$, où C est l'ensemble de concepts du métathésaurus, R est l'ensemble de types de rélations sémantiques qui peuvent exister entre les éléments de C, et E est l'ensemble de liens sémantiques qui existent dans le graphe. Nous générons alors nos données d'apprentissage à partir d'ensembles de triplets ordonnés $(h, r, t) \in E$, où $(h, r) \in C \times C$ et $r \in R$. Les valeurs possibles pour r, soit les types de regroupement sémantique du métathésaurus, sont les suivantes :

1. AQ : h peut qualifier t
2. QB : h peut être qualifié par t
3. PAR : h est un concept parent de t dans un des vocabulaires sources
4. CHD : t est un concept parent de h dans un des vocabulaires sources
5. RN : h a une définition plus étroite que t
6. RB : h a une définition plus large que t
7. SY : h et t sont synonymes

En utilisant les termes associés avec chaque concept $c \in C$, nous construisons des séquences textuelles à partir des triplets (h, r, t). Comme il peut y avoir plusieurs termes associés avec chaque concept, nous utilisons les *preferred terms* indiqués par le métathésaurus, sauf dans les cas des synonymes $r =$SY, auquel cas nous utilisons un autre terme associé avec le concept pour répresenter l'entité t.

2.2 Entraînement des modèles de langage

Pour effectuer l'entraînement par mots masqués des modèles de langue, nous séparons le corpus E3C décrit dans la section 1.3 en phrases, en coupant celles qui dépassent la longueur de séquence maximale. Pour les expériences détaillées dans ce travail, nous utilisons la configuration standard pour des modèles de type BERT ; 15% des mots masqués, taux d'apprentissage 2×10^{-5}, avec une longueur de séquence maximale de 256.

2.3 Entraînement avec une base de connaissance

À partir du métathésaurus UMLS, nous formulons deux objectifs de classification pour compléter le pré-entraînement des modèles de langue. Le premier pas pour faciliter l'intégration de ces tâches dans le processus d'entraînement est de redéfinir le vocabulaire et les tokens spéciaux utilisés par les modèles BERT. Pour les modèles que nous entraînons à partir de zéro, nous rajoutons la liste de tous les mots uniques apparaissant dans la partie française de l'UMLS (5 870 mots), en excluant ceux qui contiennent des caractères non alphabétiques, au vocabulaire initial tiré du corpus E3C. Ce vocabulaire supplementaire tiré du métathésaurus comprend les termes complexes (des concepts médicaux qui sont dénotés sous forme de groupes de mots plutôt que des mots uniques) ainsi que les termes simples. Cela permet aux transformers d'améliorer la modélisation des textes spécifiques au domaine médical et d'encoder plus directement les concepts du graphe de connaissance. Pour structurer les séquences d'entrées pour l'entraînement à base des relations sémantiques, nous rajoutons 8 tokens spéciaux aux vocabulaires des modèles ; un pour représenter chacun des types de relation qui apparaissent dans le graphe.

Classification des triplets En s'inspirant du travail de Hao *et al.* (2020), nous construisons un jeu de données pour la classification binaire des triplets (h, r, t) du graphe de connaissances comme étant vrai ou faux, où h et t sont les noms des concepts en question et r la relation entre eux. Nous tirons un échantillon des triplets du graphe comme exemples positifs. Ensuite, pour équilibrer le jeu de données avec des exemples négatifs, nous tirons des paires de concepts qui appartiennent au même groupe sémantique, mais pour lesquels une relation r n'existe pas. Afin de générer des exemples d'entraînement de faux triplets, nous utilisons deux stratégies différentes d'échantillonnage négatif.

Premièrement, pour former des exemples directement contrastés pour les relations existantes, nous échantillonnons les triples (h, r, t) où h et t appartiennent à différents groupes sémantiques et construisons des faux triplets correspondants avec le même type de relation et les mêmes catégories de groupe sémantique, c'est-à-dire $(\hat{h}, r, \hat{t}) \notin G$ où $\hat{h}$ et $\hat{t}$ appartiennent au même groupe sémantique que h et t, respectivement. Deuxièmement, afin de fournir des exemples contrastés pour les types de relation, nous échantillonnons des triplets pour lesquels h et t appartiennent au même groupe sémantique, et formons l'exemple d'entraînement négatif en changeant le type de relation r. Par souci de l'équilibre du jeu de données, les ensembles de données de classification des triples utilisés dans ce travail sont constitués de 50 % d'exemples positifs (triples réels du métathésaurus), de 25 % d'exemples générés par la première méthode d'échantillonnage négatif et du reste par la seconde. Les données d'entrée pour les transformers sont alors sous la forme `[CLS]` $w_1^h \cdots w_m^h$ `[REL]` $w_1^t \cdots w_n^t$ `[SEP]`, où `[CLS]` et `[SEP]` sont des tokens spéciaux standards pour l'encodage BERT, `[REL]` est le token spécial qui correspond à r, et les w_i^h et w_i^t sont les séquences de tokens émises par le tokenizer pour h et t respectivement. Le token `[CLS]` est transmis à une couche de classification linéaire qui produit une prédiction binaire, comme il est d'usage pour des tâches de classification avec des modèles BERT pré-entraînés.

Prédiction des entités Pour cette tâche de classification, nous complexifions le problème de classification des triplets décrit ci-dessus en le combinant avec l'objectif de prédiction des mots masqués. Les séquences d'entrées pour cette tâche ont la même structure que la précédente, mais au lieu de créer des exemples synthétiques des fausses relations et utiliser uniquement `[CLS]` pour prédire si la relation existe ou pas, nous masquons systématiquement les tokens w_i^t avec le même *mask token* utilisé pour l'entraînement à partir du texte libre. Le modèle aura alors comme objectif de prédire l'entité t d'une relation à partir de h et r. La formulation de ce dernier est partiellement basée sur celle de Meng *et al.* (2021).

Pour chacune des deux tâches décrites ci-dessus, nos expériences sont faites sur un jeu de données de 100K séquences, où les relations ont été tirées aléatoirement parmi les entrées françaises dans le tableau de relations `MRREL.RRF` du métathésaurus.

2.4 Affinage des modèles pour la reconnaissance des entités biomédicales

À partir des trois corpus d'évaluation mentionnés dans la section 1.3, nous effectuons l'affinage de bout-en-bout des modèles pour la classification des mots des cas cliniques en fonction des catégories spécifiques au domaine médical. Pour faciliter l'implémentation de ces deux tâches, nous avons pré-traité les documents pour que l'étiquetage des mots soit "plat", c'est-à-dire avoir une seule étiquette par mot. Pour effectuer ce prétraitement, nous avons utilisé directement l'ensemble des relations sémantiques UMLS pour remplacer des chevauchements par des concepts plus généraux.

Les annotations sont sous la forme BRAT (Stenetorp *et al.*, 2012), et peuvent alors être dénotées comme un ensemble $\mathcal{A} = \{a_i = (\text{CUI}^{a_i}, \text{SG}^{a_i}, s_1^{a_i}, s_2^{a_i})\} \subset C \times T \times \mathbb{N} \times \mathbb{N}$ où CUI et SG représentent l'identifiant du concept et le groupe sémantique respectivement. $s1$ et s_2 correspondent au *span* occupé par le mot dans le document en question (par souci de simplicité, nous nous limitons à la notation des spans continues, mais dans la pratique, nous traitons également des annotations discontinues). Pour aplatir les entités imbriquées, nous définissons un ensemble de concepts "de base" pour chaque groupe sémantique, en utilisant les types sémantiques, une catégorisation plus fine des concepts de l'UMLS. Ensuite, en utilisant les relations hiérarchiques (`PAR`/`CHD` et `RB`/`RN`), nous

pouvons calculer une mesure de généralité d'un concept par le nombre minimal d'étapes entre le concept et un des concepts de base. La procèdure de mise à plat des annotations $\mathcal{A}$ d'un document est alors la suivante :

1. Calculer l'ensemble (de taille n) des chevauchements entre mentions d'entités, effectivement un ensemble de combinaisons des éléments de $\mathcal{A}$:

$$\mathcal{O} = \{g_k = (a_1 \cdots a_m) : s_2^{a_1} \leq s_1^{a_2}, \ldots, s_2^{a_{m-1}} \leq s_1^{a_m}\}_{k=1}^{n}$$

2. Pour chaque élément g de $\mathcal{O}$, nous prenons ensuite l'ensemble $\phi(g) \subset C$ de concepts ayant une relation hiérarchique plus général (`PAR` ou `RB`) avec les concepts du chevauchement.
3. Les annotations imbriquées g sont alors remplacées avec l'élément de $\phi(g)$ le plus spécifique :

$$\text{argmax}_{\phi(g)} \left[\text{min}_{b \in B} d(g, b)\right]$$

où d est une fonction qui produit la "distance" hiérarchique entre deux concepts et B est l'ensemble de concepts de base. Si $\phi(g)$ est vide, nous ne gardons que l'annotation la plus courte parmi les éléments du chevauchement.

Tâche 1 : QUAERO-SG Dans cette tâche, l'objectif est de classifier les mots en fonction du groupe sémantique de l'UMLS auquel ils appartiennent. La granularité utilisée pour les annotations QUAERO nous donne dix regroupements, dont nous nous focalisons sur les suivants :

1. Désordre/trouble (`DISO`)
2. Procédure (`PROC`)
3. Structure anatomique (`ANAT`)
4. Substance chimique (`CHEM`)
5. Être vivant (`LIVB`)
6. Entité physiologique (`PHYS`)

En rajoutant une catégorie `none` pour les mots qui n'appartiennent à aucun de ces regroupements, nous obtenons alors un problème de classification avec une cardinalité de 7, pour lequel le nombre d'occurrences est affiché dans le tableau 2.

TABLE 2 – Nombre d'occurrences de chacune des catégories cibles dans la tâche QUAERO-SG.

Catégories QUAERO-SG	**train**	**dev**	**test**
DISO	1067	963	1144
PROC	741	748	719
ANAT	486	460	474
CHEM	377	395	363
LIVB	323	345	340
PHYS	177	173	157
Total	3171	2911	3040

Tâche 2 : CAS-CATEG Les cas cliniques de ce corpus sont annotés avec 14 types d'entités, 7 étant utilisées pour la tâche de reconnaissance des entités, présentées dans le tableau 3.

TABLE 3 – Nombre d'occurrences de chacune des catégories cibles dans la tâche CAS-CATEG.

Catégories CAS-CATEG	train	dev	test
sosy (signe ou symptôme)	13 977	4 940	4 297
examen	2 997	1 054	917
pathologie	2 064	460	238
traitement	1 824	859	884
moment	1 783	467	478
substance	1 517	542	346
dose	1 122	131	58
Total	25 284	8 453	7 218

3 Expériences

L'entraînement par mots masqués a été lancé à partir de deux modèles de langue pré-entraînés sur les corpus en langue française du domaine général ; les versions *base* de CamemBERT (Martin *et al.*, 2020) et FlauBERT (Le *et al.*, 2020). Nous comparons les performances de ces deux modèles avec une version du modèle DistilBERT (Sanh *et al.*, 2019) initialisé de zéro avec les mêmes paramètres d'architecture que `camembert-base`. Toutes les trois architectures ont le même nombre (12) de couches neuronales et de têtes d'attention, et la même taille de vecteurs de sortie ($d = 768$). La seule différence architecturale réside dans la valeur de la dimension intérieure : $H = 2048$ pour FlauBERT, contre $H = 3072$ pour les deux autres. Pour l'entraînement par mots masqués effectué sur le corpus E3C, nous avons entraîné chacune de ces variantes pendant 64 époques avec la méthode d'optimisation Adam avec pondération (Loshchilov & Hutter, 2017), ce que nous utilisons également pour le reste des entraînements et affinage.

Pour la tâche de classification des triplets, nous entraînons les modèles pendant 8 époques avec un taux d'apprentissage de 10^{-4}. La tâche de prédiction des entités étant plus complexe et ayant une dimension de sortie plus élevée, la décroissance de la valeur de la fonction de perte a atteint un palier autour de 24 époques, ce qui est effectivement la durée d'apprentissage pour laquelle les résultats sont présentés (le taux d'apprentissage utilisé est identique à celui de la classification des triplets).

Pour les étapes finales d'affinage pour la reconnaissance des entités, nous avons entraîné les modèles de bout-en-bout avec une couche linéaire de classification des mots pendant 4 époques sur le jeu *train*, avec les hyperparamètres suivants (choisis en fonction de la performance sur le jeu *dev*) : taux d'apprentissage 5×10^{-5}, longueur maximale des séquences 512 et une taille effective des batchs de 64. Les F-mesures (macro) sur le jeu de test pour les deux tâches de reconnaissance des entités sont indiquées dans le tableau 4, avec pour référence une comparaison avec un des modèles de langue anglaise biomédicale les plus largement utilisés, BioBERT de Lee *et al.* (2020). Dans le tableau, "MLM E3C" fait référence au pré-entraînement sur le corpus E3C, "ClfTriples" à la tâche de classification des triplets et "EntPred" à l'affinage par prédiction des entités.

Nous voyons qu'en général, l'entraînement sur le corpus E3C apporte le plus de bénéfices pour les modèles pré-entraînés. L'ajout des tâches basées sur l'UMLS apporte des améliorations variables en fonction des deux différentes tâches d'évaluation. Bien que les résultats obtenus dans ces expériences ne soient pas au niveau de l'état de l'art sur les tâches définies sur les corpus QUAERO et CAS (Hiot *et al.*, 2021; van Mulligen *et al.*, 2016), il est important de noter l'amélioration systématique de la F-mesure apportée par l'ajout des tâches d'entraînement sur les relations sémantiques UMLS,

TABLE 4 – Résultats sur les deux jeux de données test.

Modèle de base	Adaptation	QUAERO-SG	CAS-CATEG
BioBERT	-	51,8	46,1
`flaubert_base_cased`	-	55,3	46,0
	+ MLM E3C	59,4 (+4,1)	48,1 (+2,1)
	+ ClfTriples	59,7 (+0,3)	51,5 (+3,4)
	+ EntPred	64,5 (+4,8)	54,2 (+2,7)
`camembert-base`	-	57,4	46,2
	+ MLM E3C	61,8 (+4,4)	49,6 (+3,4)
	+ ClfTriples	63,1 (+1,3)	52,8 (+3,2)
	+ EntPred	**68,9 (+5,8)**	**56,6 (+3,8)**
DistilBERT *from-scratch*	+ MLM E3C	53,5	39,7
	+ ClfTriples	61,7 (+8,2)	48,0 (+8,3)
	+ EntPred	64,3 (+2,6)	53,1 (+5,1)

notamment l'objectif de classification des triplets dans le cas du modèle entraîné à partir de zéro.

4 Conclusion et Perspectives

Cet article présente une étude sur l'effet de l'augmentation des modèles de langage en utilisant une phase de pré-entraînement supplémentaire sur un graphe de connaissances médicales. Cette augmentation permet notamment d'améliorer les performances des modèles considérés sur deux tâches de reconnaissance d'entités nommées. Le code et les modèles utilisés seront mis à disposition pour permettre l'éventuelle reproduction des expériences et l'amélioration des approches.

Limitations et travaux futurs La mise en œuvre des approches proposées dans ce travail présente de nombreuses limitations qui peuvent entraver les performances. Celles-ci seront abordées dans des travaux ultérieurs. Premièrement, pour faciliter la comparaison directe des différentes phases d'entraînement, nous avons enchaîné de manière séquentielle les pré-entraînements MLM et KB, ce qui risque d'introduire un biais en faveur des objectifs les plus récents, étant donné la tendance de tels modèles de langue d'oublier des informations précédemment apprises (Korbak *et al.*, 2021). Dans les prochaines étapes de ce travail, nous comptons mélanger les séquences d'entraînement en provenance du métathésaurus avec celles provenant des corpus du texte libre, et faire apprendre les modèles avec une fonction objectif mixte (Hao *et al.*, 2020; Yao *et al.*, 2019). La continuation de ce travail comprendra également l'élargissement de l'éventail des objectifs supplémentaires ; nous étudierons l'impact de la prédiction des liens entre les concepts du graphe, ainsi que la classification des concepts par rapport à leurs catégories sémantiques dans l'ontologie UMLS. Enfin, il est important de noter la limitation des tâches d'évaluation en termes de taille des corpus et de granularité et d'exhaustivité des catégories d'entités considérées. Les travaux futurs impliqueront aussi des évaluations sur la reconnaissance des entités d'une granularité plus riche ainsi que d'autres types de tâches d'évaluation du domaine du traitement du langage clinique.

Références

ALROWILI S. & SHANKER V. (2021). BioM-Transformers : Building Large Biomedical Language Models with BERT, ALBERT and ELECTRA. p. 221–227, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.bionlp-1.24.

ALSENTZER E., MURPHY J., BOAG W., WENG W.-H., JINDI D., NAUMANN T. & MCDERMOTT M. (2019). Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, p. 72–78, Minneapolis, Minnesota, USA : Association for Computational Linguistics. DOI : 10.18653/v1/W19-1909.

AYALA SOLARES J. R., DILETTA RAIMONDI F. E., ZHU Y., RAHIMIAN F., CANOY D., TRAN J., PINHO GOMES A. C., PAYBERAH A. H., ZOTTOLI M., NAZARZADEH M., CONRAD N., RAHIMI K. & SALIMI-KHORSHIDI G. (2020). Deep learning for electronic health records : A comparative review of multiple deep neural architectures. *Journal of Biomedical Informatics*, **101**, 103337. DOI : 10.1016/j.jbi.2019.103337.

BERHE A., DRAZNIEKS G., MARTENOT V., MASDEU V., DAVY L. & ZUCKER J.-D. (2022). ALIBERT : A Pretrained language model for French biomedical text : a preprint. working paper or preprint.

BODENREIDER O. (2004). The unified medical language system (UMLS) : integrating biomedical terminology. PubMed PMID : 14681409 ; PubMed Central PMCID : PMC308795, DOI : doi : 10.1093/nar/gkh061.

CHANG D., BALAŽEVIĆ I., ALLEN C., CHAWLA D., BRANDT C. & TAYLOR A. (2020). Benchmark and Best Practices for Biomedical Knowledge Graph Embeddings. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, p. 167–176, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.bionlp-1.18.

CHENAIS G., TOUCHAIS H., AVALOS M., BOURDOIS L., REVEL P., GIL-JARDINÉ C. & LAGARDE E. (2021). Performance en classification de données textuelles des passages aux urgences des modèles BERT pour le français. In *PFIA 2021 - Journée Santé et I.A.*, Bordeaux / Virtual, France.

COPARA J., KNAFOU J., NADERI N., MORO C., RUCH P. & TEODORO D. (2020). Contextualized French Language Models for Biomedical Named Entity Recognition. In R. CARDON, N. GRABAR, C. GROUIN & T. HAMON, Éds., *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes*, p. 36–48, Nancy, France : ATALA.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DUBOIS S., ROMANO N., JUNG K., SHAH N. & KALE A. D. C. (2017). The Effectiveness of Transfer Learning in Electronic Health Records Data.

DURA B., JEAN C., TANNIER X., CALLIGER A., BEY R., NEURAZ A. & FLICOTEAUX R. (2022). Learning structures of the french clinical language : development and validation of word embedding models using 21 million clinical reports from electronic health records. DOI : 10.48550/ARXIV.2207.12940.

EL BOUKKOURI H., FERRET O., LAVERGNE T., NOJI H., ZWEIGENBAUM P. & TSUJII J. (2020). CharacterBERT : Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 6903–6915, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.609.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French Corpus with Clinical Cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, p. 122–128, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5614.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne deft 2021. *Actes de DEFT. Lille.*

HAMID O. H. (2022). From model-centric to data-centric AI : A paradigm shift or rather a complementary approach ? In *2022 8th International Conference on Information Technology Trends (ITT)*, p. 196–199. DOI : 10.1109/ITT56123.2022.9863935.

HAO B., ZHU H. & PASCHALIDIS I. (2020). Enhancing Clinical BERT Embedding using a Biomedical Knowledge Base. p. 657–661, Barcelona, Spain (Online) : International Committee on Computational Linguistics.

HIOT N., MINARD A.-L. & BADIN F. (2021). Doing@deft : utilisation de lexiques pour une classification efficace de cas cliniques. In *Actes de l'atelier Défi Fouille de Textes@TALN 2020 Classification de cas cliniques et correction automatique de copies d'étudiants. Atelier DÉfi Fouille de Textes*, p. 41–53, Lille, France : Association pour le Traitement Automatique des Langues.

KORBAK T., ELSAHAR H., KRUSZEWSKI G. & DYMETMAN M. (2021). Controlling conditional language models with distributional policy gradients. *CtrlGen @ Neural Information Processing Systems*.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020). FlauBERT : Unsupervised Language Model Pre-training for French. In *Proceedings of The 12th Language Resources and Evaluation Conference*, p. 2479–2490, Marseille, France : European Language Resources Association.

LE CLERCQ DE LANNOY T., BESANÇON R., FERRET O., TOURILLE J., BRIN-HENRY F. & VIERU B. (2022). Stratégies d'adaptation pour la reconnaissance d'entités médicales en français. In Y. ESTÈVE, T. JIMÉNEZ, T. PARCOLLET & M. ZANON BOITO, Éds., *Traitement Automatique des Langues Naturelles(TALN 2022)*, p. 215–225, Avignon, France : ATALA. HAL : hal-03701500.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics 2020*.

LI Y., RAO S., SOLARES J. R. A., HASSAINE A., RAMAKRISHNAN R., CANOY D., ZHU Y., RAHIMI K. & SALIMI-KHORSHIDI G. (2020). BEHRT : Transformer for Electronic Health Records. *Scientific Reports*, **10**, 7155. Number : 1 Publisher : Nature Publishing Group, DOI : 10.1038/s41598-020-62922-y.

LOSHCHILOV I. & HUTTER F. (2017). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics.

MENG Z., LIU F., CLARK T., SHAREGHI E. & COLLIER N. (2021). Mixture-of-Partitions : Infusing Large Biomedical Knowledge Graphs into BERT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 4672–4681, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.383.

MINARD A.-L., ZANOLI R., ALTUNA B., SPERANZA M., MAGNINI B. & LAVELLI A. (2021). European clinical case corpus. Bruno Kessler Foundation. DOI : 10.57771/DEY2-G751.

NASEEM U., BANDI A., RAZA S., RASHID J. & CHAKRAVARTHI B. R. (2022). Incorporating Medical Knowledge to Transformer-based Language Models for Medical Dialogue Generation. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, p. 110–115, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.bionlp-1.10.

NICHOLSON D. N. & GREENE C. S. (2020). Constructing knowledge graphs and their biomedical applications. *Computational and structural biotechnology journal*, **18**, 1414–1428. Place : Netherlands, DOI : 10.1016/j.csbj.2020.05.017.

NÉVÉOL A., GROUIN C., LEIXA J., ROSSET S. & ZWEIGENBAUM P. (2014). The QUAERO French Medical Corpus : A Ressource for Medical Entity Recognition and Normalization. In *Proc of BioTextMining Work*, p. 24–30.

PETERS M. E., NEUMANN M., LOGAN R., SCHWARTZ R., JOSHI V., SINGH S. & SMITH N. A. (2019). Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 43–54, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1005.

ROY A. & PAN S. (2021). Incorporating medical knowledge in BERT for clinical relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 5357–5366, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.435.

SANH V., DEBUT L., CHAUMOND J. & WOLF T. (2019). Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. *ArXiv*, **abs/1910.01108**.

STENETORP P., PYYSALO S., TOPIĆ G., OHTA T., ANANIADOU S. & TSUJII J. (2012). brat : a web-based tool for nlp-assisted text annotation.

SUSHIL M., SUSTER S. & DAELEMANS W. (2021). Are we there yet ? Exploring clinical domain knowledge of BERT models. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, p. 41–53, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.bionlp-1.5.

VAN MULLIGEN E. M., AFZAL Z., AKHONDI S. A., VO D. & KORS J. A. (2016). Erasmus mc at clef ehealth 2016 : Concept recognition and coding in french texts. In *Conference and Labs of the Evaluation Forum*.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is All you Need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 30 : Curran Associates, Inc.

WANG Y., TAFTI A., SOHN S. & ZHANG R. (2019). Applications of Natural Language Processing in Clinical Research and Practice. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Tutorials*, p. 22–25, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-5006.

YAO L., MAO C. & LUO Y. (2019). KG-BERT : BERT for knowledge graph completion. *ArXiv*, **abs/1909.03193**.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Annotation d'entités cliniques en utilisant les Larges Modèles de Langue

Simon Meoni[1, 2] Théo Ryffel[2] Éric de la Clergerie[1]
(1) Inria, Paris, France
(2) Arkhn, Paris, France
simon.meoni@arkhn.com, eric.de_la_clergerie@inria.fr

RÉSUMÉ

Dans le domaine clinique et dans d'autres domaines spécialisés, les données sont rares du fait de leur caractère confidentiel. Ce manque de données est un problème majeur lors du fine-tuning de modèles de langue. Par ailleurs, les modèles de langue de très grande taille (LLM) ont des performances prometteuses dans le domaine médical. Néanmoins, ils ne peuvent pas être utilisés directement dans les infrastructures des établissements de santé pour des raisons de confidentialité des données. Nous explorons une approche d'annotation des données d'entraînement avec des LLMs pour entraîner des modèles de moins grandes tailles mieux adaptés à notre problématique. Cette méthode donne des résultats prometteurs pour des tâches d'extraction d'information.

ABSTRACT

Annotate Clinical Data Using Large Language Model Predictions

In clinical and other specialized domains, data are scarce due to their confidential nature. This lack of data is a major problem when fine-tuning language models. Nevertheless, very large language models (LLMs) are promising for the medical domain but cannot be used directly in healthcare facilities due to data confidentiality issues. We explore an approach of annotating training data with LLMs to train smaller models more adapted to our problem. We show that this method yields promising results for information extraction tasks.

MOTS-CLÉS : Supervision Faible, Modèle de langue Large, Extraction d'information, TAL dans le domaine clinique.

KEYWORDS: Weak supervision, Large Language Model, Extraction Information Task, Clinical NLP.

1 Introduction

Les notes cliniques contiennent l'ensemble des interactions entre le patient et le personnel de santé. Les personnels de santé y indiquent notamment leurs observations et les différents actes médicaux réalisés. Malgré l'informatisation des documents cliniques, les notes doivent rester dans un format assez expressif et libre afin de faire gagner du temps au personnel soignant et de permettre la description de situations inhabituelles (Rosenbloom *et al.*, 2011). De plus, un grand nombre d'informations cruciales est contenu exclusivement dans ces notes. Selon une étude de Escudié *et al.* (2017), environ 80% des phénotypes (ensemble de caractères biologiques, physiques observables pouvant caractériser une

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 190–203.

maladie) de patients ne sont présents que dans du texte libre. Cela rend ces documents difficilement exploitables sans l'utilisation de méthodes avancées comme le deep-learning en NLP. L'utilisation de telles méthodes requiert de collecter et d'annoter un nombre important de données médicales. Cependant, Fries *et al.* (2022) met en avant le fait que les données d'apprentissage dans le domaine biomédical sont peu accessibles, peu documentées et rarement réutilisables dans un cadre commercial ou de recherche, une situation qu'il qualifie de *dataset debt*. Il montre par exemple que seulement 13% des 167 datasets analysés dans son étude sont accessibles et téléchargeables, que 22% utilisent un format standard structuré ou encore que 40% sont dans le domaine public. Ces dernières années, les larges modèles de langage (LLMs) comme GPT3 ont prouvé leur capacité à effectuer un large panel de tâches avec des performances prometteuses en contexte de zero-shot ou de few-shots. C'est une tendance très intéressante pour le développement du traitement automatique du langage naturel clinique, et ses résultats préliminaires sont prometteurs pour les tâches d'extraction d'informations (Agrawal *et al.*, 2022). Cependant, l'utilisation des LLMs soulèverait des préoccupations majeures en matière de confidentialité. En effet, il est important de garantir que le déploiement du modèle est maîtrisé et la prédiction issue du modèle doit évoluer pour s'adapter à des directives d'annotation spécifiques et changeantes. Par ailleurs, la plupart des LLMs ne sont pas disponibles librement (Scao *et al.*, 2022; Ouyang *et al.*, 2022; Thoppilan *et al.*, 2022) et à notre connaissance, seul BLOOM est open-source et déployable dans une infrastructure locale. Enfin, les ressources informatiques nécessaires pour utiliser ces modèles en inférence restent importantes, constituant de fait un obstacle de plus à leur adoption par les établissements de santé.

L'une des approches permettant de résoudre ces enjeux consiste à distiller les LLMs en modèles de taille moins conséquente via des méthodes de supervision faible. La supervision faible a récemment attiré l'attention de la communauté scientifique car elle soulage la tâche d'annotation (Lison *et al.*, 2020, 2021). En effet, cette technique consiste à annoter automatiquement des ensembles de données à l'aide de méthodes basées sur des règles, des dictionnaires ou des techniques plus avancées, puis à entraîner le modèle sur cet ensemble de données.

2 Contributions

Notre travail étudie l'utilisation de LLMs dans la technique de distillation de connaissances par le biais de techniques de supervision faible dans le domaine clinique français, en particulier dans l'extraction d'entités cliniques. Plus précisément :

— Nous montrons que la distillation est une technique compétitive par rapport aux techniques classiques de supervision faible dans le domaine clinique français ;
— Nous proposons une approche de supervision faible qui associe les annotations provenant d'extraction de dictionnaire et les prédictions de LLM pour créer un ensemble de données d'entraînement, surpassant les annotations provenant des seules prédictions d'InstructGPT-3.

3 Travaux Connexes

Supervision Faible Le *deep learning* a connu un succès remarquable dans plusieurs domaines au-delà du NLP (Zhang *et al.*, 2022). Cependant, le principal obstacle est la collecte massive de données annotées. Pour remédier à ce problème, la supervision faible remplace l'annotation de vérité

terrain par une annotation automatique basée sur des règles heuristiques ou des règles linguistiques de contrainte. Certaines techniques appelées *supervision distante* exploitent des liens sémantiques à partir de bases de connaissances ou d'ontologies (Lison *et al.*, 2021). Karamanolakis *et al.* (2021) propose une méthode d'autoapprentissage itérative pour combiner la supervision faible classique et l'inférence du modèle d'apprentissage afin d'extraire des entités non couvertes par les règles heuristiques initiales. Dans le domaine clinique, la supervision faible a déjà été utilisée pour certains cas d'usages cliniques spécifiques (Cusick *et al.*, 2021; Fries *et al.*, 2021; Wang *et al.*, 2019).

Modèles de Langage Clinique Dans un contexte spécifique au domaine clinique, certains termes spécifiques sont sous-représentés ou absents dans le domaine général. En conséquence, la communauté en NLP clinique a entraîné des modèles de langage préentraînés sur des corpus spécifiques au domaine (Alsentzer *et al.*, 2019; Lee *et al.*, 2020; Alsentzer *et al.*, 2019), tels que MIMIC-III (Johnson *et al.*, 2016) ou des résumés Pubmed. Ces modèles peuvent être entraînés à partir de poids initialisés ou à partir d'un modèle agnostique pour le spécialiser dans le domaine clinique (Gururangan *et al.*, 2020). Cependant, les gains de performance sont marginaux par rapport au modèle de langage général. La structure du texte et les abréviations présentes dans les notes cliniques ont un impact négatif sur les performances des modèles. Au lieu de préentraîner un modèle clinique spécialisé, certains travaux ont cherché à affiner des modèles LLMs génératifs de langues générales tels que la famille de modèles GPT ou T5 sur une tâche clinique. Ces modèles généraux affinés ont prouvé leur efficacité dans la réponse aux questions cliniques, la dé-identification de documents médicaux ou encore l'extraction de relations (Lehman *et al.*, 2023). Cette approche nécessite toutefois une infrastructure importante et un réaffinage régulier si la distribution des données des rapports cliniques au sein de l'établissement de santé change. Néanmoins, certains LLMs ont été préentraînés sur des notes spécifiques au domaine clinique tel que GatorTron (Yang *et al.*, 2022), BioGPT (Luo *et al.*, 2022) ou ClinicalT5 (Lu *et al.*, 2022) et ont obtenu des performances prometteuses sur plusieurs tâches.

De plus, l'apprentissage en contexte avec des LLMs de langue générale telles que InstructGPT-3 (Ouyang *et al.*, 2022), où aucun entraînement n'a été effectué, donne de bons résultats (Agrawal *et al.*, 2022; Brown *et al.*, 2020) et surpasse les modèles spécialisés de plus petites tailles sur plusieurs tâches cliniques.

Méthodes basées sur les instructions L'apprentissage basé sur les instructions pour les modèles de langage génératif traite une tâche comme un problème de modélisation de langage où un modèle de langage prédit les tokens suivant une instruction textuelle (ou *prompt*) donnée en entrée (Sainz *et al.*, 2021). Dans ce paradigme, au lieu d'affiner un modèle pour une tâche donnée (*"pré-entraînement, affinage et prédiction"*), nous voulons manipuler le comportement d'un modèle de langage pré-entraîné en utilisant un prompt approprié pour obtenir la sortie souhaitée (*"pré-entraînement, instructions et prédiction"*). L'avantage majeur de cette méthode est qu'un modèle de langage pré-entraîné de manière non supervisée peut être utilisé pour de nombreuses tâches (Liu *et al.*, 2023). De ce fait, le *prompt engineering* a été développé pour explorer la méthode d'instruction la plus adaptée appliquée à un modèle de langue pour résoudre une tâche donnée. Parmi ces méthodes, l'apprentissage en contexte (*in-context learning*) est l'une des méthodes les plus populaires pour l'extraction d'informations, la réponse aux questions ou l'analyse des sentiments. Dans le domaine clinique, certains travaux existent sur la récupération d'informations et la réponse aux questions. Le prompt contient trois composants : le modèle ou le format des exemples, l'ensemble des exemples et l'ordre des prompts tel que présenté dans la Figure 4. Le but est de fournir dans le prompt quelques exemples d'entraînement concaténés

avec l'exemple de test. Cependant, les exemples choisis, leur agencement ainsi que le format ont un impact en termes de performance (Zhao *et al.*, 2021), si bien que ces trois composants doivent être calibrés pour optimiser les performances.

4 Méthode

4.1 Annoter et distiller des connaissances via une supervision faible

Extraire des annotations à partir de la sortie LLM Notre étude s'inspire largement de la méthode développée par Agrawal *et al.* (2022). Dans ces travaux, ils évaluent la performance d'InstructGPT-3 (Ouyang *et al.*, 2022) pour des tâches de NLP clinique en anglais. InstructGPT-3 obtient des résultats prometteurs, dépassant de loin certains modèles de langues spécialisés pour plusieurs tâches d'extraction d'information. De plus, les auteurs introduisent trois nouveaux ensembles de données pour évaluer les informations cliniques dans un contexte few-shot. Enfin, un nouveau concept, nommé *requête guidée* (Figure 4) permet de pré-structurer la sortie et faciliter son exploitation via des résolveurs (ou des fonctions de *string matching*).

Notre travail se distingue par les contributions suivantes :

1. notre domaine d'étude principal est la distillation de connaissances via une supervision faible et l'amélioration de cette technique en combinant des annotations de LLMs et des méthodes basées sur de l'extraction par dictionnaire ;
2. nous appliquons nos méthodes au domaine clinique français sur lequel les données annotées sont plus rares, alors que le travail initial a été réalisé en anglais ;
3. notre travail se base sur les directives de l'ensemble de données E3C (Magnini *et al.*, 2020) pour réaliser l'extraction des entités cliniques.

Dans ce travail, InstructGPT-3 n'est pas entraîné et est utilisé tel quel ; nous nous contentons d'interroger le modèle, aucune étape d'affinage supplémentaire n'est réalisée et nous n'avons accès qu'aux paramètres d'inférence tels que la température, le top p, la pénalité de fréquence ou de présence. Nous réglons la *température* et le *top p* à 0 pour contrôler le hasard et avoir un comportement déterministe. Pour ne pas pénaliser les répétitions, nous réglons la *pénalité de présence* et la *pénalité de fréquence* à 0. Nous utilisons un modèle InstructGPT-3 (text-davinci-003) (Ouyang *et al.*, 2022) pour inférer toutes les annotations pour toutes nos expériences. Nous donnons en entrée du modèle une instruction que nous complétons par l'exemple à prédire (Figure 4). La sortie d'InstructGPT-3 est une chaîne de caractères que nous structurons afin d'aligner les entités prédites avec le texte initial (Figure 1).

La tâche consiste à annoter les mots (ou tokens) d'une phrase x avec un ensemble d'étiquettes $L := \{O, B_{\text{clin}}, I_{\text{clin}}\}$ où O dénote un mot du texte sans étiquette, B_{clin} le premier mot d'une entité clinique et I_{clin} les mots suivants selon le format *IOB* (Ramshaw & Marcus, 1995). Le but est d'analyser une phrase $x = [x_1, \ldots, x_n] \in \Sigma^*$ et d'identifier pour chaque token x_i la bonne étiquette O, B_{clin}, I_{clin} ainsi que l'offset de caractères. La prédiction du modèle est donc $\hat{y} = [y_1, y_2, \ldots, y_n] \in Y$ où une annotation y_i est définie comme $y_i := \langle x_i, s, e, l \rangle$ avec s l'offset de début, e l'offset de fin et $l \in L$.

Comme mentionné ci-dessus, une méthode basée sur des instructions nécessite de concaténer un template t avec notre phrase x pour donner notre instruction telle que $p = concat(t, x)$. Nous produisons notre sortie $o \in \Sigma^*$ à partir de notre modèle LLM Φ tel que $o = \Phi(p, \theta_h)$, où θ_h

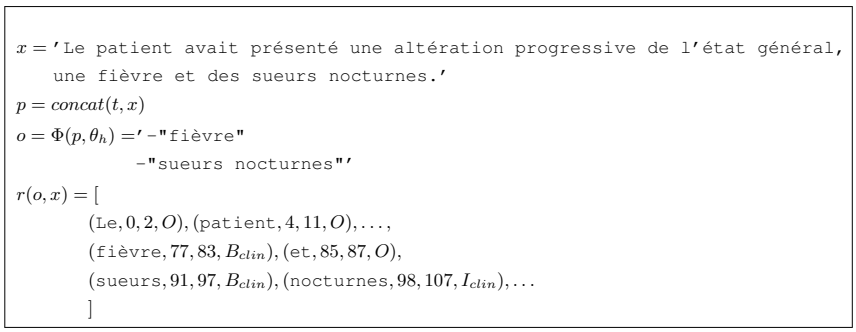


FIGURE 1 – Illustration des étapes de prédiction et de structuration de notre méthode sur un exemple. Le template t est illustré dans la Figure 4.

représente l'ensemble des hyperparamètres (*température*, *top p*, *pénalité de fréquence*, *pénalité de présence*) et o est une chaîne de caractères.

Nous structurons la sortie o en utilisant une fonction de *string matching* retournant un ensemble d'étiquettes $\hat{y} = r(o, x)$, où $r : \Sigma^* \times \Sigma^* \rightarrow Y$ est le résolveur appliquant la fonction de *string matching*.

Distillation de connaissances via la supervision faible Enfin, les annotations générées via la prédiction d'InstructGPT-3 sont utilisées comme ensemble d'entraînement pour affiner un modèle de langage plus petit pour la tâche cible.

4.2 Instructions

Nous fournissons des instructions de mise en contexte avec trois exemples de données annotées. Nous sélectionnons deux exemples avec plusieurs entités cliniques extraites et un autre sans entités. Ce paramétrage permet de fournir une diversité sémantique d'entités à InstructGPT-3 (Figure 4).

Après plusieurs essais empiriques et analyses qualitatives, nous ajoutons un exemple sans entités cliniques pour éviter trop de faux positifs extraits par le modèle. Nous insérons dans les prompts un nombre réduit de mots clés associés à la définition de l'E3C des entités cliniques. De plus, nous ajoutons les tokens de *requête guidée* pour expliciter la structure de la réponse afin de faciliter le *parsing* de la sortie du LLM (Agrawal *et al.*, 2022).

5 Expériences

5.1 Jeu de données

Nous utilisons le corpus multilingue (anglais, basque, espagnol, français, italien) E3C (Magnini *et al.*, 2020) pour nos expériences, qui se compose de deux types d'annotations : temporelles et entités cliniques. Comme mentionné précédemment, notre objectif est d'extraire des entités cliniques en français. Dans E3C, l'une des tâches supplémentaires est de lier les entités extraites avec des entrées du métathésaurus de l'UMLS. Toutefois, nous nous concentrons sur l'extraction des entités cliniques.

Le jeu de données E3C est organisé en trois couches, chacune correspondant à un ensemble de données annotées d'une certaine manière :

— La première couche (appelée **layer 1**) consiste en une annotation manuelle complète ;
— La deuxième couche (**layer 2**) consiste en une annotation semi-automatique réalisée via une extraction par dictionnaire, qui contient des termes de l'UMLS et des termes extraits de la **layer 1**. Un sous-ensemble de cette couche (environ 10%) a été corrigé manuellement [1] (**layer 2 validate**) et est également utilisé séparément dans notre étude .
— La troisième couche (**layer 3**) est une couche non annotée que non-utilisée dans nos travaux.

Nous avons sélectionné le sous-ensemble français d'E3C comprenant des données annotées manuellement et automatiquement pour nos expériences de supervision faible. Cependant, le jeu de données a une quantité limitée de données dans ses différentes couches. Pour pallier ce problème, nous avons divisé le **layer 2** en utilisant une validation croisée à 5 partitions (folds).

5.2 Protocole Expérimental

Nous menons des expériences sur des tâches d'extraction d'entités cliniques. Pour toutes nos expériences, nous utilisons *camembert-base* (Martin *et al.*, 2019) comme modèle étudiant pour l'étape de distillation des connaissances via de la supervision faible. Nous testons nos modèles sur le **layer 1**. Les différents corpus d'entraînement utilisés dans les différentes configurations seront issus du **layer 2** et du **layer 2 validate**. Nous utilisons soit les annotations initiales, soit les annotations inférées par InstructGPT-3, soit encore un mélange des deux. Nous menons nos expériences en utilisant quatre configurations de jeux de données différentes pour l'affinage de *camembert-base* :

— **Configuration A** : nous utilisons le **layer 2** comme jeu d'entraînement et comparons deux modèles camemBERT entraînés respectivement sur des annotations d'extraction de dictionnaire et des annotations prédites par InstructGPT-3 ;
— **Configuration B** : nous utilisons le **layer 2 validate** comme jeu d'entraînement et comparons deux modèles camemBERT entraîné respectivement sur des annotations corrigées manuellement et des annotations prédites par InstructGPT-3 ;
— **Configuration C** : nous nous basons sur la **Configuration A** mais nous conservons les annotations de base du **layer 2 validate** pour les deux modèles. Ainsi, une petite partie (environ 10%) des annotations prédites par InstructGPT-3 et des annotations d'extraction de dictionnaire sont remplacées par des annotations manuelles ;
— **Configuration D** : nous utilisons le **layer 2** comme jeu d'entraînement mais mixons un ratio r d'annotations prédites par InstructGPT-3 avec $(1 - r)$ d'annotations prédites par InstructGPT-3. Dans ce contexte, nous testons et comparons des modèles entraînés pour divers valeurs de ratio r.

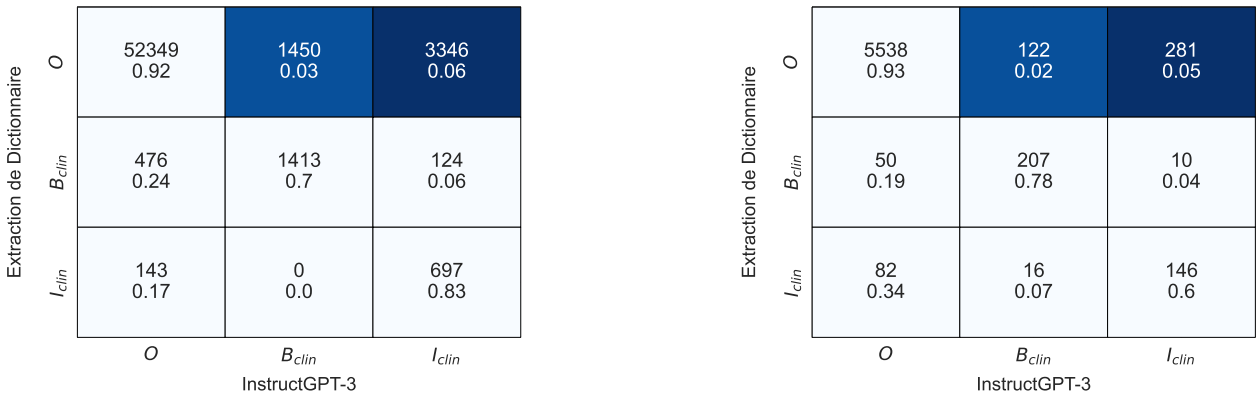
5.3 Résultats et discussion

Analyse des prédictions d'InstructGPT-3 Nous notons qu'InstructGPT-3 extrait presque deux fois plus d'entités (Table 1). Cette tendance est plus importante dans le **layer 2**, tandis que l'écart est réduit dans le **layer 2 validate** ; cela est certainement dû à la validation humaine sur cette couche. Cette différence pourrait s'expliquer par le fait qu'InstructGPT-3 n'a pas accès aux guidelines ; l'instruction reproduite en Figure 4 mentionne des "*disorders*", "*diseases*", ou "*symptoms*" à extraire :

1. ce sous-ensemble validée (**layer 2 validate**) n'a cependant pas été corrigé de la même manière que le **layer 1**

Layer	Méthodes	#Phrases	#Tokens	B_{clin}	I_{clin}
Layer 1	Dictionary Extraction	1109	28744	1258	1203
	InstructGPT-3			1398	2028
Layer 2	Dictionary Extraction	2389	59998	2013	840
	InstructGPT-3			2863	4167
Layer 2 Validate	Manual Extraction	293	6452	267	244
	InstructGPT-3			345	437

TABLE 1 – Nombre de tokens annotés par type d'annotation pour chaque couche



(a) **layer 2** (b) **layer 2 validate**

FIGURE 2 – Matrices de confusion pour le **layer 2** et le **layer 2 validate** montrant les relations entre les annotations par dictionnaire et via InstructGPT-3. Par exemple dans le **layer 2 validate**, l'intersection B_{clin} (en abscisse) à O (en ordonnée) indique qu'InstructGPT a identifié 122 tokens O comme B_{clin}en se basant sur l'extraction par dictionnaire.

ceci est moins restrictif que l'annotation de la guideline d'E3C. Les matrices de confusion (Figure 2) soulignent cette tendance. Néanmoins, les deux méthodes d'extraction ont presque étiqueté le même nombre de tokens en tant que O (0.93), ce qui confirme que l'annotation d'InstructGPT-3 est cohérente avec la tâche d'extraction. Nous notons que seulement 0.04 de B_{clin} sont annotés en tant que token I_{clin}, ce qui signifie qu'InstructGPT-3 discrimine bien avec la signification des tokens de début (B_{clin}) et des tokens internes (I_{clin}). Nous soulignons le fait est qu'InstructGPT-3 considère certains tokens O comme des tokens internes : les entités extraites sont plus longues que celles extraites par dictionnaire. D'où le plus grand nombre de tokens extrait par InstructGPT-3.

Évaluation de la distillation des connaissances Nous comparons dans le Table 2 le modèle distillé via les annotations d'InstructGPT-3 (*Modèle Distillé*) et le modèle InstructGPT-3 sur le **layer 1**. Les deux modèles ont presque les mêmes performances, mais le modèle distillé a une meilleure précision, tandis qu'InstructGPT est plus performant en termes de rappel. Nous pouvons conclure que l'utilisation d'un modèle distillé est pertinent vis à vis de cette tâche.

Configuration A Si nous comparons le score F1 global (Table 3a), le modèle distillé via les annotations d'InstructGPT-3 (*Modèle Distillé*) est plus performant que le modèle (*Modèle par Supervision Faible*) formé avec la supervision faible par extraction par dictionnaire. Le *Modèle par Supervision*

Modèles	F1-Score	Précision	Rappel
Modèle Distillé	$0.74 \pm 6e^{-3}$	**0.72** $\pm$ 0.01	$0.78 \pm 6e^{-3}$
InstructGPT-3	0.74	0.70	**0.81**

TABLE 2 – Comparaison d'InstructGPT-3 avec un modèle distillé via de la supervision faible sur le **layer 2** d'E3C. Les scores sont calculés sur le **layer 1**.

Faible a un meilleur score F1 avec la classe B_{clin} , mais le rappel et le score F1 de I_{clin} sont relativement plus faibles.

Le *Modèle par Supervision Faible* a tendance à mieux extraire les termes cliniques composés d'un seul mot. Le *Modèle Distillé* a un meilleur rappel et reconnaît plus facilement les termes cliniques à plusieurs mots. Cependant, cette flexibilité est contrebalancée par la détection de faux positifs, ce qui fait baisser la précision.

Configuration B La quantité de tokens annotés est relativement faible par rapport au **layer 2** (Table 1). Cela nuit au résultat (Tableau 3b) du *Modèle par Supervision Faible* (0.57 avec le **layer 2 validate** contre 0.69 avec le **layer 2**). Contrairement au *Modèle Distillé*, où les performances sont relativement équivalentes par rapport au **layer 2**. Cependant, la stabilité du *Modèle Distillé* entre les 5 folds est affectée (0.01 avec le **layer 2 validate** vs. $6e^{-3}$ avec le **layer 2**). Le même problème de stabilité des folds est présent pour *Modèle par Supervision Faible*, ce qui peut être dû à la quantité relativement faible de tokens dans le jeu d'entraînement.

Configuration C Les résultats (Table 3c) montrent des performances légèrement meilleures pour les deux modèles. Nous remarquons que le *Modèle par Supervision Faible* a un meilleur rappel pour I_{clin} que dans la **Configuration A** (*Modèle par Supervision Faible* entraîné avec seulement des annotations de supervision faible) : respectivement 0.35 contre 0.24. Le *Modèle Distillé* présente une meilleure précision pour le B_{clin} et le I_{clin} par rapport au *Modèle Distillé* entraîné avec le **layer 2**. En conclusion, le mélange de jeux de données comportant une faible proportion d'annotations manuelles corrige la tendance du *Modèle par Supervision Faible* à ignorer les entités composées de plusieurs mots et celle du *Modèle Distillé* à inclure des mots supplémentaires aux entités présentes dans le corpus de test, notamment celles composées de peu de mots.

Configuration D Les résultats indiquent que l'utilisation qu'un ratio de 0.5 entre l'annotation prédite par InstructGPT-3 et l'annotation extraction par dictionnaire permet d'améliorer le F1-Score de 0.03 par rapport à l'utilisation d'un ensemble de données entièrement annoté avec l'annotation prédite par InstructGPT-3 (c'est-à-dire lorsque le rapport est de 1). En outre, l'incorporation d'une proportion d'annotations prédites par InstructGPT-3 au-dessus de 0.5 permet une meilleure stabilité entre les différentes folds. En conclusion, la combinaison de ces deux méthodes d'annotation disparates en termes de rappel et de précision permet d'obtenir un bon équilibre et d'augmenter le F1-Score.

Synthèse entre les différentes configurations La Table 4 montre que la configuration D est celle qui a obtenu légèrement un meilleur F1-score (0.76). Le mélange d'annotations par extraction de dictionnaire et InstructGPT-3 obtenu avec $r = 0.5$ permet de réduire l'écart entre les Rappel

Annotations	extraction par dictionnaire	InstructGPT-3
F1-score	0.69 ± 0.04	**0.74** $\pm 6e^{-3}$
B_{clin}	F : **0.76** ± 0.02 P : **0.83** $\pm 2e^{-3}$ R : $0.70 \pm 4e^{-2}$	F : $0.73 \pm 8e^{-3}$ P : 0.68 ± 0.02 R : **0.80** ± 0.01
I_{clin}	F : 0.34 ± 0.10 P : **0.69** ± 0.01 R : 0.24 ± 0.08	F : **0.53** ± 0.01 P : 0.41 ± 0.01 R : **0.74** ± 0.01

(a) **Configuration A**

Annotations	extraction par dictionnaire	InstructGPT-3
F1-score	0.57 ± 0.02	**0.74** ± 0.01
B_{clin}	F : 0.72 ± 0.03 P : **0.65** ± 0.07 R : 0.82 ± 0.04	F : **0.74** ± 0.03 P : 0.64 ± 0.06 R : **0.87** ± 0.02
I_{clin}	F : 0.02 ± 0.04 P : 0.33 ± 0.30 R : 0.01 ± 0.02	F : **0.53** ± 0.01 P : **0.42** ± 0.03 R : **0.71** ± 0.03

(b) **Configuration B**

Annotations	extraction par dictionnaire	InstructGPT-3
F1-score	0.73 ± 0.01	**0.75** $\pm 5e^{-3}$
B_{clin}	F : **0.78** $\pm 7e^{-3}$ P : $0.84 \pm 4e^{-3}$ R : **0.72** ± 0.01	F : $0.77 \pm 5e^{-3}$ P : **0.85** ± 0.01 R : 0.71 ± 0.01
I_{clin}	F : 0.46 ± 0.03 P : **0.66** ± 0.04 R : 0.35 ± 0.04	F : **0.50** ± 0.01 P : 0.64 ± 0.02 R : **0.42** ± 0.02

(c) **Configuration C**

TABLE 3 – Performances obtenues en utilisant les deux paramétrages avec les deux jeux d'annotations. La ligne F1-score dénote les macro-F1-scores agrégés des labels O, B_{clin}, I_{clin}.

Configuration	F1-Score	Précision	Rappel
A	0.74 ± 0.01	0.69 ± 0.01	0.83 ± 0.01
B	0.74 ± 0.02	0.69 ± 0.03	**0.84** ± 0.01
C	0.75 ± 0.00	**0.82** ± 0.01	0.71 ± 0.01
D r=0.5	**0.76** ± 0.01	0.73 ± 0.02	0.81 ± 0.03

TABLE 4 – Performances pour les configurations décrites en section 5.2.

(0.81) et la Précision (0.73). Ceci nous permet d'affirmer que les deux méthodes peuvent être complémentaires : une combinaison des deux permet d'obtenir une diversité d'annotations plus importante que l'utilisation seule d'une des deux méthodes.

6 Limitations

Les limites de notre étude est la taille petite du corpus de test, ce qui peut avoir un impact sur la généralisation de nos résultats. Nous avons limité notre travail à l'extraction d'entités cliniques ; dans des travaux futurs, nous expérimenterons d'autres tâches en utilisant la couche de temporalité E3C pour couvrir une tâche de reconnaissance d'entités nommées et d'extraction de relations.

Enfin, les guidelines E3C ont été conçues pour l'extraction d'entités cliniques associables aux concepts de l'UMLS. Après la première étape d'annotation manuelle, une partie de l'étendue des entités a été modifiée pour correspondre le plus possible aux concepts sémantiques trouvés dans l'UMLS (Magnini *et al.*, 2020). Ces biais induisent des difficultés supplémentaires quant aux résultats de la prédiction des modèles sur le **layer 1**.

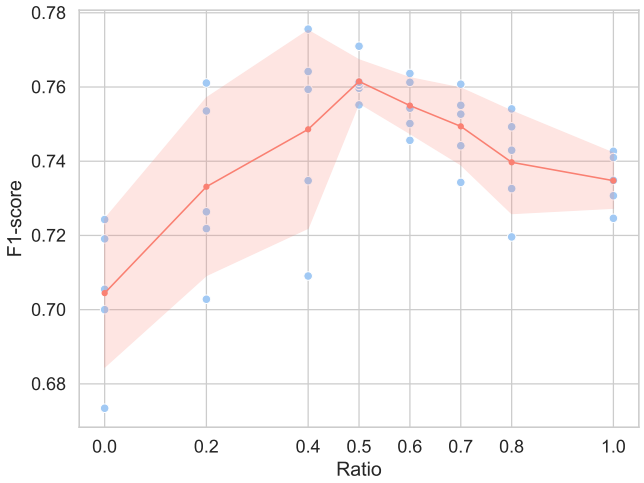


FIGURE 3 – La figure représente un graphique avec le F1-Score en ordonnée et le ratio r d'annotations par dictionnaire et d'annotations via InstructGPT-3 en abscisse comme précisé dans la **Configuration D** (Section 5.2). $r = 0$ indique l'utilisation exclusive d'annotations de dictionnaire, et $r = 1$ d'annotations de InstructGPT-3 exclusivement. Chaque point bleu représente une expérience différente avec un ensemble de données différent. Un point orange représente la moyenne des expériences pour r donné. La bande en orange clair représente l'écart-type pour r donné.

7 Conclusion

Nos résultats montrent que l'approche de distillation de connaissances avec InstructGPT-3 surpasse l'approche d'extraction de dictionnaire pour extraire les entités cliniques. Nous avons montré que mélanger ces approches pour construire un ensemble de données d'entraînement apporte de la diversité aux annotations et améliore les performances du modèle entraîné.

L'approche de faible supervision avec les LLMs est prometteuse pour créer un ensemble de données d'entraînement. Cela réduit le coût d'annotation et, en même temps, concentre l'annotation manuelle sur l'ensemble de test, qui est l'un enjeux majeurs des domaines à enjeux élevés comme la santé.

Dans les travaux futurs, nous aimerions tester cette solution sur plusieurs langues proposées dans E3C et étendre notre travail à d'autres tâches. Nous voulons également combiner plusieurs prédictions de différents grands modèles de langage pour obtenir plus de diversité dans les annotations.

De plus, nous avons l'intention d'étudier des techniques plus avancées pour combiner les différentes annotations en incorporant des mesures de confiance provenant des différentes prédictions ou bien des mesures de performance (telles que le rappel et la précision) pour décider quel type d'annotations (B_{clin} ou I_{clin}) conserver pour chaque méthode de prédiction.

Enfin, l'adaptation de CoT ou de connaissances générées (Wei *et al.*, 2022; Cobbe *et al.*, 2021) pour l'extraction d'entités cliniques pourrait être bénéfique pour améliorer la précision des LLMs. Nous pourrions créer un prompt où les différentes étapes d'annotation sont présentées à travers différents exemples. À chaque étape d'annotation, nous décrivons une instruction précise et son résultat. Par exemple, nous pouvons rédiger dans une instruction, les trois étapes d'annotation E3C pour encourager le LLM à être plus proche des guidelines d'identification de termes cliniques.

Références

AGRAWAL M., HEGSELMANN S., LANG H., KIM Y. & SONTAG D. (2022). Large Language Models are Few-Shot Clinical Information Extractors.

ALSENTZER E., MURPHY J. R., BOAG W., WENG W.-H., JIN D., NAUMANN T. & MCDERMOTT M. B. A. (2019). Publicly Available Clinical BERT Embeddings. DOI : 10.48550/arxiv.1904.03323.

BROWN T. B., MANN B., RYDER N., SUBBIAH M., KAPLAN J., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., AGARWAL S., HERBERT-VOSS A., KRUEGER G., HENIGHAN T., CHILD R., RAMESH A., ZIEGLER D. M., WU J., WINTER C., HESSE C., CHEN M., SIGLER E., LITWIN M., GRAY S., CHESS B., CLARK J., BERNER C., MCCANDLISH S., RADFORD A., SUTSKEVER I. & OPENAI D. A. (2020). Language Models are Few-Shot Learners.

COBBE K., KOSARAJU V., BAVARIAN M., CHEN M., JUN H., KAISER L., PLAPPERT M., TWOREK J., HILTON J., NAKANO R., HESSE C. & SCHULMAN J. (2021). Training Verifiers to Solve Math Word Problems. DOI : 10.48550/arxiv.2110.14168.

CUSICK M., ADEKKANATTU P., CAMPION T. R., SHOLLE E. T., MYERS A., BANERJEE S., ALEXOPOULOS G., WANG Y. & PATHAK J. (2021). Using weak supervision and deep learning to classify clinical notes for identification of current suicidal ideation. *Journal of Psychiatric Research*, **136**, 95–102. DOI : 10.1016/J.JPSYCHIRES.2021.01.052.

ESCUDIÉ J.-B., RANCE B., MALAMUT G., KHATER S., BURGUN A., CELLIER C. & JANNOT A.-S. (2017). A novel data-driven workflow combining literature and electronic health records to estimate comorbidities burden for a specific disease : a case study on autoimmune comorbidities in patients with celiac disease. *BMC Medical Informatics and Decision Making*, **17**(1), 140. DOI : 10.1186/s12911-017-0537-yï.

FRIES J. A., SEELAM N., ALTAY G., WEBER L., KANG M., DATTA D., SU R., GARDA S., WANG B., OTT S., SAMWALD M. & KUSA W. (2022). Dataset Debt in Biomedical Language Modeling. *Workshop on Challenges & Perspectives in Creating Large Language Models*, **5**, 137–145.

FRIES J. A., STEINBERG E., KHATTAR S., FLEMING S. L., POSADA J., CALLAHAN A. & SHAH N. H. (2021). Ontology-driven weak supervision for clinical entity classification in electronic health records. *Nature Communications*, **12**(1). DOI : 10.1038/s41467-021-22328-4.

GURURANGAN S., MARASOVÍCMARASOVÍC A., SWAYAMDIPTA S., LO K., BELTAGY I., DOWNEY D. & SMITH N. (2020). Don't Stop Pretraining : Adapt Language Models to Domains and Tasks. *58th Annual Meeting of the Association for Computational Linguistics*, p. 8342–8360.

JOHNSON A. E., POLLARD T. J., SHEN L., LEHMAN L. W. H., FENG M., GHASSEMI M., MOODY B., SZOLOVITS P., ANTHONY CELI L. & MARK R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific data*, **3**. DOI : 10.1038/SDATA.2016.35.

KARAMANOLAKIS G., MUKHERJEE S., ZHENG G. & AWADALLAH A. H. (2021). Self-Training with Weak Supervision. p. 845–863. DOI : 10.18653/v1/2021.naacl-main.66.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240. DOI : 10.1093/bioinformatics/btz682.

LEHMAN E., HERNANDEZ E., MAHAJAN D., WULFF J., SMITH M. J., ZIEGLER Z., NADLER D., SZOLOVITS P., JOHNSON A. & ALSENTZER E. (2023). Do We Still Need Clinical Language Models ?

LISON P., BARNES J. & HUBIN A. (2021). skweak : Weak Supervision Made Easy for NLP.

LISON P., BARNES J., HUBIN A. & TOUILEB S. (2020). Named Entity Recognition without Labelled Data : A Weak Supervision Approach. p. 1518–1533. DOI : 10.18653/V1/2020.ACL-MAIN.139.

LIU P., YUAN W., JIANG Z., HAYASHI H., NEUBIG G., FU J., YUAN W., JIANG Z., HAYASHI H., NEUBIG G. & FU J. (2023). Pre-train, Prompt, and Predict : A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, **55**(9), 1–35. DOI : 10.1145/3560815.

LU Q., DOU D. & NGUYEN T. H. (2022). ClinicalT5 : A Generative Language Model for Clinical Text.

LUO R., SUN L., XIA Y., QIN T., ZHANG S., POON H. & LIU T.-Y. (2022). BioGPT : Generative Pre-trained Transformer for Biomedical Text Generation and Mining. *Briefings in bioinformatics*, **23**(6). DOI : 10.1093/bib/bbac409.

MAGNINI B., ALTUNA B., LAVELLI A., SPERANZA M. & ZANOLI R. (2020). The E3C Project : Collection and Annotation of a Multilingual Corpus of Clinical Cases. *CEUR Workshop Proceedings*, **2769**. DOI : 10.4000/BOOKS.AACCADEMIA.8663.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE V., SEDDAH D. & SAGOT B. (2019). CamemBERT : a Tasty French Language Model. p. 7203–7219. DOI : 10.18653/v1/2020.acl-main.645.

OUYANG L., WU J., JIANG X., ALMEIDA D., WAINWRIGHT C. L., MISHKIN P., ZHANG C., AGARWAL S., SLAMA K., RAY A., SCHULMAN J., HILTON J., KELTON F., MILLER L., SIMENS M., ASKELL A., WELINDER P., CHRISTIANO P., LEIKE J. & LOWE R. (2022). Training language models to follow instructions with human feedback. DOI : 10.48550/arxiv.2203.02155.

RAMSHAW L. A. & MARCUS M. P. (1995). Text Chunking using Transformation-Based Learning. p. 157–176. DOI : 10.48550/arxiv.cmp-lg/9505040.

ROSENBLOOM S. T., DENNY J. C., XU H., LORENZI N., STEAD W. W. & JOHNSON K. B. (2011). Data from clinical notes : A perspective on the tension between structure and flexible documentation. *Journal of the American Medical Informatics Association*, **18**(2), 181–186. DOI : 10.1136/JAMIA.2010.007237.

SAINZ O., DE LACALLE O. L., LABAKA G., BARRENA A. & AGIRRE E. (2021). Label Verbalization and Entailment for Effective Zero- and Few-Shot Relation Extraction. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, p. 1199–1212. DOI : 10.48550/arxiv.2109.03659.

SCAO T. L., FAN A., AKIKI C., PAVLICK E. & ET AL. (2022). BLOOM : A 176B-Parameter Open-Access Multilingual Language Model. DOI : 10.48550/arxiv.2211.05100.

THOPPILAN R., DE FREITAS D., HALL J., SHAZEER N., KULSHRESHTHA A., CHENG H.-T., JIN A., BOS T., BAKER L., DU Y., LI Y., HUAIXIU H. L., ZHENG S., GHAFOURI A., MENEGALI M., HUANG Y., KRIKUN M., LEPIKHIN D., QIN J., CHEN D., XU Y., CHEN Z., ROBERTS A., BOSMA M., ZHAO V., ZHOU Y., CHANG C.-C., KRIVOKON I., RUSCH W., PICKETT M., SRINIVASAN P., MAN L., MEIER-HELLSTERN K., RINGEL M., TULSEE M., RENELITO D., SANTOS D., DUKE T., SORAKER J., ZEVENBERGEN B., PRABHAKARAN V., DIAZ M., HUTCHINSON B., OLSON K., MOLINA A., HOFFMAN-JOHN E., LEE J., AROYO L., RAJAKUMAR R., BUTRYNA A., LAMM M., KUZMINA V., FENTON J., COHEN A., BERNSTEIN R., KURZWEIL R., AGUERA-ARCAS B., CUI C., CROAK M., CHI E. & LE GOOGLE Q. (2022). LaMDA : Language Models for Dialog Applications.

WANG Y., SOHN S., LIU S., SHEN F., WANG L., ATKINSON E. J., AMIN S. & LIU H. (2019). A clinical text classification paradigm using weak supervision and deep representation. *BMC Medical Informatics and Decision Making*, **19**(1), 1–13. DOI : 10.1186/S12911-018-0723-6/FIGURES/4.

WEI J., WANG X., SCHUURMANS D., BOSMA M., ICHTER B., XIA F., CHI E., LE Q. & ZHOU D. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models.

YANG X., CHEN A., POURNEJATIAN N., SHIN H. C., SMITH K. E., PARISIEN C., COMPAS C., MARTIN C., COSTA A. B., FLORES M. G., ZHANG Y., MAGOC T., HARLE C. A., LIPORI G., MITCHELL D. A., HOGAN W. R., SHENKMAN E. A., BIAN J. & WU Y. (2022). A large language model for electronic health records. *npj Digital Medicine*, **5**(1). DOI : 10.1038/s41746-022-00742-2.

ZHANG J., HSIEH C.-Y., YU Y., ZHANG C. & RATNER A. (2022). A Survey on Programmatic Weak Supervision. DOI : 10.48550/arxiv.2202.05433.

ZHAO T. Z., WALLACE E., FENG S., KLEIN D. & SINGH S. (2021). Calibrate Before Use : Improving Few-Shot Performance of Language Models. DOI : 10.48550/arxiv.2102.09690.

A Appendice

```
Input: L'évolution était marquée deux mois plus tard, par
l'apparition de placards angiomateux au
niveau de l'avant bras droit, [...]
extract the exact match of disorders, diseases or
symptoms mentioned in the text or
return None if there is no clinical entity:

- "placards angiomateux"
- "lymphoedème"
- "lésions"

Input: De façon concomitante, le patient avait présenté une
altération progressive de l'état général,
une fièvre et des sueurs nocturnes
extract the exact match of disorders, diseases
or symptoms mentioned in the text or
return None if there is no clinical entity:

- "altération progressive de l' état général"
- "fièvre"
- "sueurs nocturnes"

Input: La vitesse de sédimentation était à 35mm à
la première heure, la protéine C réactive
était négative et
la Ferritinémie était à 900µg/l (soit à 4 fois la normale).
extract the exact match of disorders, diseases
or symptoms mentioned in the text or
return None if there is no clinical entity:

- "None"

Input: L'interrogatoire n'a retrouvé aucun antécédent
pathologique en particulier la notion d'éruption cutanée,
de troubles du transit, d'ictère,
d'épisode infectieux respiratoire ou de vaccination récente.
extract the exact match of disorders, diseases
or symptoms mentioned in the text or
return None if there is no clinical entity:
- "
```

FIGURE 4 – Un exemple du texte d'instruction utilisé dans notre expérience. Les exemples formatés sont présentés en bleu, tandis que l'exemple à prédire est présentés en orange. Les instructions sont présentées en violet, et la *requête guidée*, telle qu'utilisée dans Agrawal *et al.* (2022), sont présentées en vert.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Classification de tweets en situation d'urgence pour la gestion de crise

Romain Meunier[1] Leila Moudjari[1] Farah Benamara[1] Véronique Moriceau[1]
Alda Mari[2] Patricia Stolf[1]
[1]IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France
[1]prenom.nom@irit.fr
[2]IJN, CNRS/ENS/EHESS, Université Paris Saclay
[2]alda.mari@ens.fr

RÉSUMÉ

Le traitement de données provenant de réseaux sociaux en temps réel est devenu un outil attractif dans les situations d'urgence, mais la surcharge d'informations reste un défi à relever. Dans cet article, nous présentons un nouveau jeu de données en français annoté manuellement pour la gestion de crise. Nous testons également plusieurs modèles d'apprentissage automatique pour classer des tweets en fonction de leur pertinence, de l'urgence et de l'intention qu'ils véhiculent afin d'aider au mieux les services de secours durant les crises selon des méthodes d'évaluation spécifique à la gestion de crise. Nous évaluons également nos modèles lorsqu'ils sont confrontés à de nouvelles crises ou même de nouveaux types de crises, avec des résultats encourageants.

ABSTRACT

Tweets Classification in Emergency Situations for Crises Management

The processing of real-time social media data has become an attractive tool in emergency situations. However, information overload remains a challenge. In this paper, we present a new dataset in French, annotated for crisis management. Relying on evaluation methods specifically designed to crisis management, we evaluate several deep learning models for classification of tweets based on their relevance, urgency and intention they convey to help emergency services during crises. We also evaluate these models on new crises or new types of crises showing interesting results.

MOTS-CLÉS : Gestion de crise, Réseaux sociaux, Apprentissage multi-tâches, Portabilité.

KEYWORDS: Crisis management, Social Media, Multi-task Learning, Portability.

1 Motivations

Les réseaux sociaux ont changé la façon dont les gens interagissent, communiquent ou ont accès à l'information partout dans le monde. Leur utilisation est largement répandue chez les internautes, les institutions publiques et les entreprises pour informer à propos d'événements ou de produits ou encore, et sans être exhaustifs, partager des opinions. Tous les domaines de la vie quotidienne sont concernés, y compris la gestion de crise qui nous intéresse ici. En effet, les catastrophes naturelles représentent une menace permanente pour les populations, les infrastructures et les ressources naturelles. La gestion efficace de ces crises nécessite des outils et des approches innovants pour aider les équipes

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 204–216.

d'intervention à réagir rapidement et à prendre des décisions éclairées (Vieweg *et al.*, 2014; Olteanu *et al.*, 2015; Palen & Liu, 2007). Dans ce contexte, les réseaux sociaux, en particulier Twitter, offrent une source d'information riche et en temps réel sur les événements en cours (Reuter *et al.*, 2018). Par exemple, plus de 17 000 tweets ont été postés pendant l'incendie de Notre Dame en 2019. Lors du séisme de 2023 en Turquie et en Syrie, des victimes piégées sous les décombres ont appelé à l'aide en publiant des messages sur Twitter (Toraman *et al.*, 2023).

Ainsi, Twitter peut être utilisé pour obtenir des informations cruciales en situation de crise comme l'état des infrastructures ou bien le niveau de préparation de la population face à la crise.

L'une des principales caractéristiques des tweets postés en période de crise est la variété des informations véhiculées qui entraîne des différences de traitement par les services de secours. Par exemple, dans le tweet (a) [1] l'auteur signale un danger qui approche et le message doit donc être traité en priorité. Le tweet (b) exprime quant à lui une critique envers le système de gestion de la crise. Même si ce tweet peut être pris en compte afin d'améliorer le service, il n'a pas de caractère urgent et ne nécessite donc pas d'intervention des secours. Enfin, l'exemple (c) est un avertissement officiel. Il doit être pris en compte par les secours afin qu'ils puissent se préparer au mieux à la catastrophe arrivant.

(a) Le feu de Landiras (au départ à 40km) s'approche de chez moi. Encore 2 villages et c'est à nous d'évacuer. Ce soir on sent vachement le brûlé. On ne panique pas mais le stress monte. Vais mal dormir.

(b) Pourquoi ne pas faire intervenir l'Armée ? Inondations à Villeneuve S/Lot en 59 : les militaires nous apportaient à manger. L'Etat abandonne les sinistrés français !

(c) ALERTE MÉTÉO : Vigilance #orange "orages, pluie-inondation" et #jaune "vagues-submersion". Soyez prudents

Ces dernières années, la littérature sur la gestion de crise a connu une expansion rapide. Elle se focalise surtout sur l'anglais, et malgré la quantité importante de jeux de données élaborés pour la gestion de crise (Imran *et al.*, 2016; McCreadie *et al.*, 2019, 2020; Alam *et al.*, 2018), les ressources en français sont encore rares, peu disponibles et très souvent limitées à une seule crise en particulier. Dans ce cadre, nos objectifs sont multiples. Il s'agit : (1) de détecter automatiquement les tweets pertinents lors d'une crise ; (2) parmi les tweets utiles/pertinents, d'identifier ceux qui sont urgents (au sens où ils doivent être traités de façon urgente par les services de secours) et (3) de caractériser le type d'information véhiculée (conseil, critique, etc.), que nous appellerons *intention*. D'un point de vue linguistique, ce terme est employé pour décrire les postures qu'un individu peut prendre eu égard à une action à entreprendre (Grano, 2017; Giannakidou & Mari, 2021). Dans cette étude, nous employons le terme pour désigner les domaines d'actionnabilité même, à savoir les domaines au sein desquels une action pourrait être entreprise (par exemple les dégâts matériels ou humains pour solliciter de l'aide). Afin de traiter ces trois points, nous proposons :

— Un corpus de tweets en français annoté à la fois selon l'utilité, l'urgence et l'intention du message [2], et qui se caractérise par sa diversité en termes aussi bien de types que de nombre de crises (crises naturelles ou non, prévisibles ou soudaines),

— *Une approche supervisée* pour la classification de tweets en situation de crise. Nous évaluons sur ce corpus un ensemble de modèles qui ont fait leurs preuves dans ce cadre, avec en particulier

1. Les exemples de tweets sont issus de notre corpus.
2. Le corpus est disponible sur demande.

une approche multi-tâches. Notre objectif n'est pas de proposer de nouveaux modèles mais d'évaluer les performances et la portabilité de modèles existants quand ils sont confrontés à des crises de types différents ou à des crises inconnues,

— *Une analyse d'erreurs* montrant les limites des approches proposées et qui permet de proposer des orientations pour améliorer les modèles dans ce domaine.

Dans cet article, nous commençons par présenter les travaux existants sur la gestion de crise pour les médias sociaux, ainsi que les corpus existants en français. La section 3 présente notre corpus annoté. La section 4 présente les modèles et les expérimentations de classification que nous avons menées sur ce corpus, ainsi qu'une analyse d'erreurs permettant d'identifier les principaux défis à relever et pistes d'amélioration pour le futur dans ce domaine. Enfin, nous concluons avec quelques perspectives pour des travaux futurs en section 5.

2 État de l'art

De nombreux travaux d'analyse des tweets en cas de crises ont émergé dans différentes langues, majoritairement l'anglais, que ce soit pendant ou après une crise (Cameron *et al.*, 2012; Imran *et al.*, 2013; McCreadie *et al.*, 2019; Kayi *et al.*, 2020; Seeberger & Riedhammer, 2022; Imran *et al.*, 2014). L'objectif est de proposer un système de classification des messages selon : (1) l'utilité (le message est-il utile ou non pour les services de secours ?), (2) l'urgence (le message est-il urgent ou non pour les services de secours ? Éventuellement, quel degré d'urgence ?) et (3) les intentions véhiculées (déclaration de dégâts, avertissement, critique, etc.). D'autres classifications ont également été proposées comme la détection de messages postés par des témoins directs ou indirects de la crise (Zahra *et al.*, 2020). Des campagnes d'évaluation ont été proposées dans ce domaine, notamment TREC-IS (Incident Streams track) [3] associé à TREC 2019 et 2020 (McCreadie *et al.*, 2019, 2020) et plus récemment CrisisFACTS2022 [4] pour la génération de résumé de situations de crises. Les participants de TREC-IS ont pour objectif le développement de systèmes de veille en temps réel capables de suivre l'évolution d'incidents tels que des catastrophes naturelles, des incidents terroristes ou des crises de santé publique à partir de flux de données textuelles en ligne, tels que des flux Twitter ou des flux de nouvelles en langue anglaise. Les meilleurs systèmes soumis utilisent des approches neuronales. Par exemple, Wang *et al.* (2021) et Dusart *et al.* (2021) utilisent une variation du modèle BERT, qui a été pré-entrainé sur des tweets relatifs à des crises dans une approche multi-tâche. Globalement, les résultats montrent que la détection de l'intention reste la plus complexe en raison du déséquilibre des données d'apprentissage (les messages urgents sont souvent minoritaires, par exemple de l'ordre de 3,87 % pour les dégâts matériels et 1,93 % pour les dégâts humains).

En ce qui concerne les ressources en langue française, on peut citer le projet SURICATE-Nat [5] qui vise à exploiter Twitter pour être informé rapidement de séismes ou d'inondations ayant lieu en France, ou encore le cadre proposé par (Interdonato *et al.*, 2018) pour extraire et regrouper automatiquement des données relatives à une crise sans recourir à une annotation manuelle ou à des catégories prédéfinies. Cependant, les données de cette étude ne sont pas disponibles. Kozlowski *et al.* (2020) ont proposé le premier corpus en français, composé d'environ 13 000 tweets concernant des crises écologiques (catastrophes naturelles) et annotés manuellement selon trois niveaux d'information :

3. https://www.dcs.gla.ac.uk/~richardm/TREC_IS/
4. https://crisisfacts.github.io/
5. http://www.suricatenat.fr/Suricate-Nat/

l'utilité, l'urgence et les intentions. Plus récemment, Diwersy *et al.* (2022) ont utilisé un corpus de discours publics pour analyser les changements dans l'utilisation des mots et expressions pendant la crise sanitaire de la COVID-19 en France, en utilisant des méthodes phonético-textométriques. Caillaut *et al.* (2022) ont construit automatiquement un corpus de 6 023 documents contenant 304 826 entités liées en utilisant la Wikipédia française afin de géolocaliser les publications des réseaux sociaux en temps réel lors de catastrophes naturelles. Enfin, le projet CrisisNLP [6] rassemble une communauté de chercheurs autour de la création de ressources et d'outils de traitement de texte pour l'analyse de données de médias sociaux pendant les crises. Le projet a produit des corpus dans plusieurs langues, dont le français (17 329 tweets relatifs à un événement de glissement de terrain survenu en 2015).

En gestion de crises, les méthodes les plus anciennes utilisées pour classifier les messages s'appuyaient sur l'apprentissage automatique avec notamment l'utilisation de Random Forest (Breiman, 2001) ou des réseaux bayésiens (Schneider, 2003). Les approches utilisées plus récemment sont de deux types : celles qui utilisent des modèles d'apprentissage profond comme les réseaux neuronaux convolutifs (Nguyen *et al.*, 2017) ou les réseaux de neurones récurrents (Alharbi & Lee, 2019), et celles qui utilisent les transformeurs (Zahera *et al.*, 2019) avec des modèles tels que BERT (Devlin *et al.*, 2019). La campagne d'évaluation TREC-IS (Incident Stream) (McCreadie *et al.*, 2020) donne un apercu des méthodes actuelles pour l'identification de tweets en situation de crise et lors de l'édition 2021, les transformeurs sont les modèles qui ont obtenu les meilleurs résultats [7].

Pour conclure, un défi important est la capacité des systèmes à identifier des messages urgents pour des crises nouvelles non présentes lors de l'entraînement. Or, d'après le rapport du GIEC 2022 (De Pryck, 2022), le changement climatique va entraîner de nouveaux types de crises et il faut donc avoir un modèle capable de se préparer à l'inconnu pour un déploiement chez les acteurs de cellules de crises. Dans cet article, nous nous intéressons pour la première fois à ce défi sur des données en français en évaluant la portabilité de nos modèles à différents types de crises.

3 Corpus de tweets en français annoté pour la gestion de crise

Le seul corpus pour la gestion de crise existant pour le français et disponible pour la communauté est celui proposé par (Kozlowski *et al.*, 2020) [8]. Ce corpus a l'inconvénient d'être composé quasi exclusivement de tweets en rapport avec des crises météorologiques prévisibles (tempête, ouragan, inondation). Nous avons choisi d'étendre ce corpus à d'autres types de crises, à savoir des crises non prévisibles afin de tester la portabilité des modèles sur ces nouveaux types. Pour cela, nous avons collecté des tweets en français portant sur deux nouveaux types de crise (incendie et attaque terroriste) tout en suivant la même méthodologie de collecte utilisée dans (Kozlowski *et al.*, 2020), c'est-à-dire collecter des tweets postés entre 24h avant la crise et 72h [9] après en utilisant des mots-clés représentatifs de la crise (par exemple, "incendie", "forêt" et "gironde" pour les incendies dans les Landes). Ces nouvelles crises ont la particularité d'être des crises soudaines : l'attaque terroriste de Trèbes en 2018, l'incendie de l'usine Lubrizol en 2019, l'incendie de Notre Dame en 2019, l'explosion/incendie d'un immeuble à Sanary en 2021, les incendies dans les Landes (Landiras

6. https://crisisnlp.qcri.org/
7. https://trecis.github.io/
8. https://github.com/DiegoKoz/french_ecological_crisis
9. Pour le cas des crises s'étalant sur plusieurs jours comme les incendies, la fin de la crise est considérée comme étant le début de résolution de la crise. Par exemple, dans le cas des incendies, il s'agit de l'extinction des premiers feux.

Tweet	Utilité	Urgence	Intention
Quatre départements du sud-est placés en **vigilance** orange aux pluies et inondations	utile	urgent	avert. conseil
Un **bébé** se **noie** dans l'inondation de l'appartement de ses parents	utile	urgent	dégâts humains
Un **immeuble s'effondre** en plein centre ville de Marseille http ://bit.ly/2zxte0S	utile	urgent	dégâts matériels
Inondations de l'Aude : la **solidarité** syndicale s'organise	utile	non urgent	soutien
Étonnant les feux d'artifice dans le #Gard, région d' #Uzes alors que ça brûle en #Gironde #14Juillet2022	utile	non urgent	critiques
#Inondations dans l'Aude : les maires et les députés au plus près des sinistrés	utile	non urgent	autres messages
tu mets le feu	non utile	non utile	non utile

TABLE 1 – Extrait du corpus avec les annotations associées.

et La Teste de Buch) en 2022. À ces crises, nous avons aussi ajouté celle de l'effondrement de 2 immeubles à Lille survenu en 2022 (le type "effondrement" n'étant représenté que par une seule crise dans le corpus initial de (Kozlowski *et al.*, 2020)). Ces tweets ont ensuite été annotés par 2 paires d'annotateurs selon le même schéma d'annotation que celui proposé par (Kozlowski *et al.*, 2020) (cf. Figure 1). Selon ce schéma, les messages ont d'abord été annotés en utilité, puis en urgence et enfin en intentions pour les messages jugés utiles. Les annotateurs se sont d'abord entraînés sur des tweets issus du corpus de (Kozlowski *et al.*, 2020) puis ont annoté les mêmes tweets que ceux utilisés pour calculer l'accord-interannotateur dans (Kozlowski *et al.*, 2020). Les kappas étant alors similaires (autour de 0,70 pour l'utilité, 0,67 pour l'urgence et 0,65 pour l'intention), les annotateurs ont annoté le corpus en entier.

La table 1 donne quelques exemples du corpus ainsi que les labels qui leurs sont associés. Les mots en gras sont des mots qui ont été décisifs pour décider quel label associer à quel message.

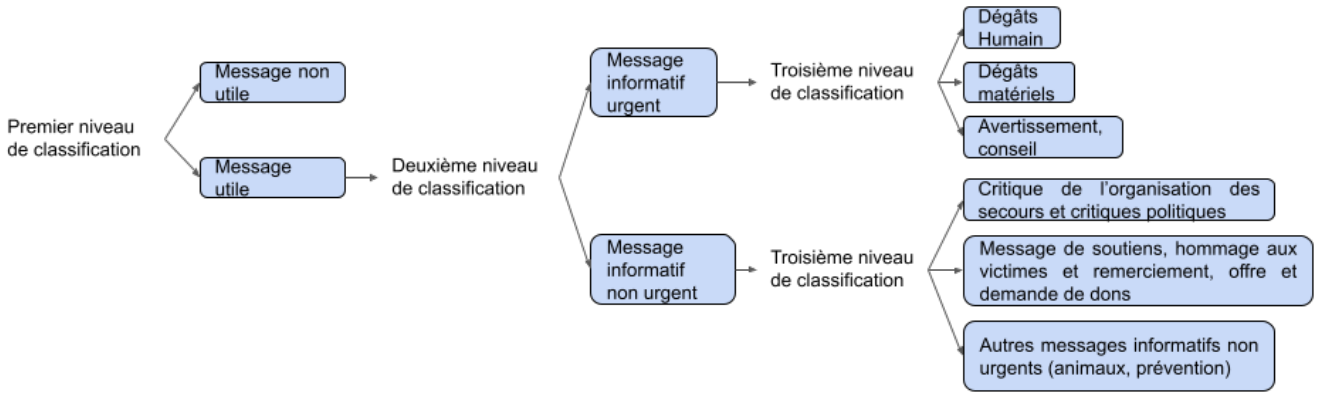


FIGURE 1 – Typologie des messages pour les crises écologiques (Kozlowski *et al.*, 2020).

La table 2 présente la distribution par classe pour toutes les crises disponibles. Il faut noter que certaines crises (incendie de l'usine Lubrizol, incendie d'un immeuble à Sanary, incendie de Notre-Dame et attaque de Trèbes) ont été annotées uniquement selon le premier et le second niveau de classification. La collecte de nouvelles crises a permis d'augmenter le corpus de (Kozlowski *et al.*, 2020) de 52,70 %. Les crises prévisibles (inondations, tempêtes, ouragans) sont les plus représentées avec 12 112 messages contre 7 483 messages pour les crises soudaines (effondrements, incendies, attaque). La collecte de nouvelles données a donc permis de rééquilibrer ces deux types de crises. Logiquement, on remarque aussi que pour les crises soudaines, il y a moins de messages de type AVERTISSEMENT-CONSEIL que pour les crises prévisibles.

Enfin, on note que le corpus est déséquilibré : la classe MESSAGE NON UTILE représente 57,96 % du corpus alors que la classe MESSAGE URGENT 20,26 %. Ceci est dû en grande partie à la méthode de collecte des tweets puisque des tweets postés 24 heures avant les crises ont été récupérés et

	non utile	utile-urgent			utile-non urgent			Total
		avert. conseil	dégâts humains	dégâts matériels	soutien	autres messages	critiques	
Corpus Kozlowski et al.								
Inondation Aude	1 065	150	34	157	157	184	26	**1 773**
Inondation Autre	993	292	35	111	231	16	19	**1 697**
Inondation Corse	468	51	58	12	52	66	13	**720**
Tempête Guadeloupe	612	91	0	2	3	10	2	**720**
Tempête Bruno	586	107	5	11	2	9	0	**720**
Tempête Susanna	484	129	11	38	4	54	0	**720**
Tempête Ulrika	650	47	2	18	0	4	0	**721**
Tempête Réunion	587	56	5	9	12	46	5	**720**
Tempête Fionn	552	138	6	10	0	8	6	**720**
Tempête Egon	609	66	1	35	0	10	0	**721**
Tempête Eleanor	590	82	22	19	1	6	0	**720**
Ouragan Harvey	628	78	10	2	1	1	0	**720**
Ouragan Irma	790	121	47	55	199	199	29	**1 440**
Effondrement Marseille	627	9	24	11	11	19	19	**720**
Nouvelles crises soudaines								
Effondrement Lille	320	2	39	27	12	117	32	**549**
Incendie Gironde Landes	1 394	51	23	93	317	380	165	**2 423**
Incendie Lubrizol	137	583			627			**1 347**
Incendie Sanary	6	363			164			**533**
Incendie Notre-Dame	86	224			209			**519**
Attaque Trèbes	174	398			810			**1 382**
Total	**11 358**	**3 970**			**4 257**			**19 595**

TABLE 2 – Statistiques du corpus annoté.

sont donc très majoritairement non pertinents. En ce qui concerne les intentions, les classes sont aussi déséquilibrées : la classe DÉGÂTS HUMAINS représente seulement 1,93 % du corpus avec 306 messages tandis que la classe AVERTISSEMENT-CONSEIL est constituée de 1 470 tweets soit 9,88 % du corpus. La gestion de crise à partir de tweets s'apparente donc à un problème de détection de signaux faibles.

4 Modèles proposés et résultats

Nous avons évalué ce corpus avec un ensemble de modèles qui ont fait leurs preuves dans ce cadre, avec en particulier une approche multi-tâche. Ont été testés différents modèles tels que BERT, CamemBERT mais nous ne présentons ici que les modèles qui ont obtenu les meilleurs résultats, à savoir :

— **FlauBERT**$_{Fine-Tuned}$: il s'agit d'un modèle FlauBERT (Le *et al.*, 2019) pré-entraîné sur 358 834 tweets non annotés du corpus de (Kozlowski *et al.*, 2020) et qui a montré de meilleures performances comparé au modèle FlauBERT (Le *et al.*, 2019) pre-entraîné sur du texte français provenant de différentes sources (par exemple Wikipédia et livres). Pour entraîner le modèle, nous avons utilisé un optimisateur Adam, avec un taux d'apprentissage de 2^{-5} sur 4 epochs.

— **FlauBERT**$_{Fine-Tuned+MultiTask}$: nous avons également entraîné notre modèle FlauBERT$_{Fine-Tuned}$ avec une architecture multi-tâches. L'objectif est de partager les connais-

sances entre les trois classifieurs de tâche (utilité, urgence et intention) avec un entraînement conjoint. Dans le modèle FlauBERT$_{Fine-Tuned+MultiTask}$, chaque classifieur est entraîné pour une tâche spécifique. Tous les classifieurs partagent les mêmes couches basses (qui sont des couches de **FlauBERT**$_{Fine-Tuned}$) sauf la dernière qui est spécifique à la tâche. Pour entraîner le modèle, nous avons utilisé un optimisateur Adam, avec un taux d'apprentissage de 2^{-5} sur 4 epochs.

Le corpus étant déséquilibré, nous utilisons la fonction de perte *focal loss* (Lin *et al.*, 2016) plus appropriée pour traiter les classes déséquilibrées [10]. Un prétraitement des messages est aussi effectué : les nombres sont remplacés par un token "nombre", les mentions sont supprimées.

4.1 Protocole d'évaluation

Chaque modèle est évalué pour chaque tâche : (1) la tâche d'utilité qui est un problème binaire (MESSAGE UTILE vs. MESSAGE NON UTILE), (2) la tâche d'urgence qui est un problème ternaire (MESSAGE NON UTILE vs. MESSAGE NON URGENT vs. MESSAGE URGENT), et (3) la tâche de détection d'intention (7 classes). Pour la tâche d'intention, les tweets non annotés en intention ne sont pas pris en compte.

Notre objectif étant d'évaluer les performances des différents modèles lorsqu'ils sont confrontés à différents types de crise, nous les avons testés dans les configurations suivantes :

1. **Généralisation** : Afin d'évaluer la portabilité des modèles aux crises soudaines, nous les avons entraînés sur le corpus initial de (Kozlowski *et al.*, 2020) et les avons testés sur les nouvelles crises que nous avons ajoutées.
2. **Hors-événement (Out-OF-Event)** : Inspirée de (Nguyen *et al.*, 2016) cette évaluation consiste à entraîner un modèle sur un ensemble de crises (par exemple, Ouragan Irma et Tempête Egon) et tester sur d'autres crises du même type (par exemple, Ouragan Harvey et Tempête Fionn). Pour constituer notre corpus d'entraînement, nous avons appliqué la méthode utilisée pour TREC-IS 2018 (McCreadie *et al.*, 2019) et sélectionné pour chaque type de crise, celle qui possède le plus de messages afin d'avoir la meilleure couverture pour chaque type de crise. Ainsi, les crises qui font partie du jeu d'entraînement sont : Inondation Aude, Tempête Egon, Ouragan Irma, Effondrement Marseille, Incendie Gironde Landes et Attaque Trèbes. Ceci donne un total de 8 459 messages pour les tâches de détection d'utilité et d'urgence, et 7 077 messages pour la tâche d'intention. Le jeu de test est composé de 11 136 messages pour les tâches d'utilité et d'urgence et 8 737 messages pour la tâche d'intention et est composé des crises non présentes dans le jeu d'entraînement.
3. **Hors-Type (Out-OF-Type)** : Cette configuration consiste à entraîner le modèle sur des types de crises spécifiques (par exemple, les inondations et les ouragans) puis à les tester sur un autre type de crise (par exemple, les incendies). Le but est de vérifier si le modèle peut s'adapter à de nouvelles crises qu'il ne connaît pas, et donc à de potentielles crises futures. Dans le cas d'une évaluation Hors-Type, pour avoir un résultat représentatif, il faut tester sur chaque type de crise individuellement puis faire la moyenne des F-scores obtenus pour chaque crise. Les résultats présentés ici sont calculés de cette façon. Nous avons fait une expérimentation Hors-Type

10. Nous avons également testé la Weigthed cross-entropy et la cross-entropy mais les résultats étaient moins bons.

sur 5 types de crises (Incendie, Inondation, Ouragan, Tempête, Effondrement) [11]. La méthode d'évaluation hors-type est une évaluation récente et encore peu exploitée : Kersten *et al.* (2019) et Algiriyage *et al.* (2021) utilisent cette stratégie mais sur un nombre de types de crises très réduit. Bourgon *et al.* (2022) ont également utilisé une évaluation similaire. Cependant, la plupart de ces travaux se concentrent uniquement sur la classification des messages en utilité et urgence. Nous allons plus loin ici en abordant, en plus, la tâche plus complexe de classification en intention.

Il est à noter que l'évaluation Hors-Événement est la méthode d'évaluation standard en gestion de crise. Nous proposons dans cet article deux protocoles supplémentaires afin d'évaluer la portabilité des modèles vers de nouvelles crises.

4.2 Résultats

La table 3 présente les résultats obtenus par les deux meilleurs modèles sur les tâches de prédiction de l'utilité, de l'urgence et des intentions dans la configuration *Généralisation*. À titre de comparaison, les macro F1-scores obtenus par (Kozlowski *et al.*, 2020) sur le corpus initial sont de 85,3 pour la tâche d'utilité, de 76,7 pour l'urgence et 65,4 pour les intentions avec le même modèle FlauBERT$_{Fine-Tuned}$ (respectivement, 85,4, 77,5 et 64,0 pour le modèle FlauBERT$_{Fine-Tuned+MultiTask}$). Sans surprise, les résultats obtenus quand les modèles sont testés sur de nouveaux types de crise sont moins bons mais sont tout de même honorables. Globalement, le modèle FlauBERT$_{Fine-Tuned+MultiTask}$ obtient de meilleurs résultats.

Modèles	Corpus de test		**utilité**	**urgence**	**intention**
FlauBERT$_{Fine-Tuned}$	Kozlowski	Macro F1	85,3	76,7	65,4
FlauBERT$_{Fine-Tuned}$	Corpus augmenté	Précision	59,84	**55,87**	**49,82**
		Rappel	60,23	55,88	49,07
		Macro F1	59,92	**55,42**	44,21
FlauBERT$_{Fine-Tuned+MultiTask}$	Corpus augmenté	Précision	**65,34**	54,00	45,12
		Rappel	**65,62**	**62,52**	**54,57**
		Macro F1	**64,66**	55,25	**46,42**

TABLE 3 – Résultats pour l'expérimentation de généralisation.

Les tables 4 et 5 présentent les résultats obtenus sur les 3 tâches dans les configurations *Hors-Événement* et *Hors-Type*. Là aussi, le modèle multi-tâches obtient globalement de meilleurs résultats.

Les résultats dans les configurations *Hors-Événement* et *Hors-Type* sont relativement similaires pour le modèle FlauBERT$_{Fine-Tuned+MultiTask}$ alors que FlauBERT$_{Fine-Tuned}$ performe mieux dans l'expérimentation Hors-Événement. Il faut noter que les tailles des corpus d'entraînement respectifs sont très différentes. En effet, le jeu d'entraînement de l'expérimentation *Hors-Événement* contient 8 459 messages alors que la taille du corpus d'entraînement *Hors-Type* varie selon le type de crise testé : en moyenne, un jeu d'entraînement pour une expérimentation *Hors-Type* contient 15 017 messages, soit 1,77 fois plus de données d'entraînement que pour l'expérimentation *Hors-Événement*.

11. Afin de garantir un jeu de test homogène pour les 3 tâches qui nous concernent, les données sur l'attaque terroriste de Trèbes n'ont pas été considérées car pas annotées en intention. Pour les données sur les incendies, seule la crise des incendies dans les Landes a été conservée, car elle a la seule annotée dans les 3 tâches.

Modèles		utilité	urgence	intention
FlauBERT$_{Fine-Tuned}$	Précision	65,69	58,85	**56,04**
	Rappel	65,67	60,15	47,37
	Macro F1-Score	65,68	59,39	**50,31**
FlauBERT$_{Fine-Tuned+MultiTask}$	Précision	**70,58**	**62,72**	52,31
	Rappel	**70,28**	**64,47**	**48,92**
	Macro F1-Score	**72,53**	**63,13**	49,65

TABLE 4 – Résultats pour l'expérimentation Hors-Événement.

Modèles		utilité	urgence	intention
FlauBERT$_{Fine-Tuned}$	Précision	73,09	**64,19**	51,26
	Rappel	**75,06**	**65,12**	46,06
	Macro F1-Score	73,19	**63,75**	45,87
FlauBERT$_{Fine-Tuned+MultiTask}$	Précision	**74,88**	63,73	**53,13**
	Rappel	74,58	64,37	**51,49**
	Macro F1-Score	**73,98**	63,04	**50,20**

TABLE 5 – Résultats pour l'expérimentation Hors-Type.

On en déduit donc que, étant donné que FlauBERT$_{Fine-Tuned+MultiTask}$ s'entraîner sur les 3 taches simultanément, il apprend plus vite et a donc besoin de moins de données pour être performant.

Pour l'expérimentation *Hors-Type*, les moins bons résultats sont obtenus sur les crises soudaines : ainsi avec FlauBERT$_{Fine-Tuned}$, on obtient un F1-score de 31,33 quand on teste sur les crises de type effondrement, 40,24 pour les incendies alors qu'on obtient un F1-score de 56,00 quand on teste sur les ouragans. Pour comparer avec les résultats de (Kozlowski *et al.*, 2020), leurs expérimentations *Hors-Type* sur le corpus initial (en testant les modèles sur l'effondrement de Marseille) avaient obtenu un F1-Score moyen de 42,30 avec FlauBERT$_{Fine-Tuned+MultiTask}$, contre 50,20 sur notre corpus augmenté. On peut donc penser que notre nouveau corpus permet de mieux appréhender l'arrivée d'un nouveau type de crise.

Si l'on compare maintenant les performances des deux modèles testés, FlauBERT$_{Fine-Tuned+MultiTask}$ est le plus performant pour la tâche de détection d'utilité. Cependant, en fonction du type d'expérimentation, FlauBERT$_{Fine-Tuned}$, peut être plus efficace, par exemple sur la tâche de prédiction de l'urgence en configuration *Hors-Type*. Pour la tâche de prédiction des intentions, qui est la tâche la plus complexe car 7 classes possibles à prédire, FlauBERT$_{Fine-Tuned+MultiTask}$ obtient de meilleurs résultats dans les configurations *Généralisation* et *Hors-Type*. Il s'agit de deux tâches dont la difficulté est que certains types de crises du jeu de test ne sont pas connus du modèle car absents du jeu d'apprentissage.

4.3 Analyse d'erreurs

Pour analyser les erreurs de classification, nous nous intéressons à la tâche la plus complexe, à savoir la détection des intentions dans un cadre *Hors-Type*. Nous analysons ici en détail les résultats du modèle FlauBERT$_{Fine-Tuned+MultiTask}$ qui a obtenu les meilleurs résultats sur cette tâche.

Dans la table 6, on remarque que les classes pour lesquelles le modèle a les moins bons résultats sont les classes AUTRES MESSAGES et CRITIQUES, tandis que la classe où le modèle est le plus performant est MESSAGES NON UTILES, celle-ci étant la classe comportant le plus de données d'entraînement. De plus, étant donné qu'il s'agit d'une classe commune aux tâches d'intention, d'urgence et d'utilité, le modèle multi-tâches s'entraîne 3 fois plus sur cette classe. Il est donc normal que le modèle soit le plus performant sur cette classe.

	non utile	**avert. conseil**	**autres messages**	**dégâts matériels**	**Soutien**	**dégâts humains**	**critiques**	**Moyenne**
Précision	81,94	46,50	33,14	53,54	59,25	55,07	42,47	53,13
Rappel	83,56	51,34	28,29	52,15	64,76	52,46	27,87	**51,49**

TABLE 6 – Résultats du modèle FlauBERT$_{Fine-Tuned+MultiTask}$ dans la configuration *Hors-Type*.

La classe AUTRES MESSAGES est celle pour laquelle les résultats sont les moins bons. Cependant, d'un point de vue applicatif, on peut relativiser ces résultats car d'après (Kozlowski *et al.*, 2020), "la catégorie AUTRES MESSAGES n'a pas d'impact immédiat sur les actions à mettre en oeuvre mais contribuent à informer les personnes sur la situation. Ils regroupent : (i) les messages à propos d'animaux, (ii) les messages qui ont pour but de donner des informations via un lien, des photos ou des vidéos et (iii) des messages de prévention qui donnent des conseils sur la crise en cours".

Texte	**Label**	**Prédiction**
(1) Abritez vous réfugiez vous sur les étages ne sortez pas Signalez votre présence au numero numero Aude Inondation Catastrophe VigilanceRouge	Autres messages	Avert-Conseil
(2) C est nul Dans l Aude numero morts numero disparus le chef des pompiers de Carcassonne a constaté avec colère un afflux de touristes voyeurs dans les communes sinistrées J invite les gens à venir armés de pelles et de raclettes pas de leur portable pour aider les habitants	Autres messages	Critiques
(3) Inondations dans l Aude les maires et les députés au plus près des sinistrés	Autres messages	Soutiens
(4) Certaines personnes commencent déjà à s installer pour dormir la journée a été longue Trebes Aude	Autres messages	Message non utile
(5) INONDATIONS À Carcasonne l eau continue de monter la préfecture invite les habitants qui vivent au bord de l Aude à monter dans les étages L hôpital est inondé intemperies	Avert-Conseil	Dégâts Matériels
(6) Si c est aussi rapide et efficace qu à St Martin et à St Barth je souhaite bon courage aux pauvres victimes de ces inondations	Critiques	Soutiens
(7) Courage à nos compatriotes victimes des intempéries dans l Aude Malheureusement plusieurs victimes à déplorer Nos services de sécurité et de secours sont remarquables	Dégâts Humains	Soutiens
(8) Inondations l aéroclub de narbonne est en danger de mort intemperies	Dégâts Matériels	Dégâts Humains
(9) C est la tempête dehors je sais pas comment j ai survécu	Message Non Utile	Autres messages
(10) Comme c est désolant de voir des immeubles s effondrer suite à un gros manque d entretien De tout coeur avec les habitants de cette avenue et ceux qui y travaillent tout les jours Aux collectivités de réagir maintenant	Soutiens	Critiques

TABLE 7 – Exemples d'erreurs de classification.

La table 7 présente quelques exemples d'erreur de classification. Dans le premier message, on constate que le cas (iii) de la définition ci-dessus pose problème car il est très proche de la définition de la catégorie AVERTISSEMENT - CONSEIL.

En dehors de ce type d'erreur spécifique à la catégorie AUTRES MESSAGES, les erreurs dans les autres catégories sont plutôt similaires. Par exemple, pour le deuxième message, l'erreur de classification s'explique par le fait que l'intention du message est de critiquer la population et pas le système de secours, ce qui le différencie d'un message de type CRITIQUES. Mais comme le terme "chefs des pompiers" est utilisé, cela peut induire le modèle en erreur.

Il existe aussi le problème des intentions multiples : certains messages véhiculent bien l'intention indiquée dans le label mais l'intention prédite par le modèle est tout aussi vraie (exemples (3) et (7)).

Le manque de contexte peut aussi expliquer certaines erreurs de classification (exemples (4) et (9)).

Certaines erreurs peuvent aussi s'expliquer par des usages figuratifs (métaphores, ironie, etc.) : par exemple dans le message (8), l'utilisation du mot "mort" va entraîner une classification en DÉGÂTS HUMAINS alors qu'ici c'est un aéroclub qui est concerné. Le message (6) est quant à lui porteur d'ironie, couramment utilisée pour exprimer des critiques mais très difficile à détecter automatiquement (Zeng & Li, 2022). Ceci explique peut être pourquoi la classe CRITIQUES a les deuxièmes moins bons résultats.

5 Conclusions et perspectives

Nous avons présenté dans cet article un nouveau corpus en français annoté pour la gestion de crise. Ce corpus est varié en types et en nombre de crises : crises prévisibles (tempête, ouragan, inondation) et crises soudaines (effondrement, incendie, attaque terroriste). Cette diversité permet aux intervenants de se préparer à faire face aux situations imprévues et à mettre en place des stratégies de gestion de crise adaptées. Nous avons mené des expérimentations pour évaluer l'efficacité de modèles d'apprentissage supervisé pour la détection automatique des tweets pertinents et urgents liés à une crise. Nous avons montré que les modèles obtenaient des résultats encourageants lorsqu'ils sont testés sur de nouvelles crises ou même de nouveaux types de crises. Au vu des résultats, nous constatons que le problème reste encore déséquilibré en termes de classes comme en termes de représentativité des crises. Aussi, nous envisageons d'augmenter automatiquement le corpus sur les classes et les crises les moins représentées. Une autre piste envisagée est de chercher d'autres sources d'information liées aux crises et de les combiner avec les tweets dans une perspective de détection multi-modale.

Remerciements

Ce travail a été réalisé dans le cadre du projet CNRS-prématuration INTACT impliquant l'Institut de Recherche en Informatique de Toulouse (IRIT) et l'Institut Jean Nicod (IJN).

Références

ALAM F., OFLI F. & IMRAN M. (2018). CrisisMMD : Multimodal Twitter Datasets from Natural Disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*.

ALGIRIYAGE N., SAMPATH R., PRASANNA R., DOYLE E. E., STOCK K. & JOHNSTON D. (2021). Identifying disaster-related tweets : a large-scale detection model comparison. In *Social Media in Crises and Conflicts, Proceedings of the 18th ISCRAM Conference*, p. 731–743.

ALHARBI A. & LEE M. (2019). Crisis detection from arabic tweets. In *Proceedings of the 3rd workshop on Arabic corpus linguistics*, p. 72–79.

BOURGON N., BENAMARA F., MARI A., MORICEAU V., CHEVALIER G. & LEYGUE L. (2022). Are Sudden Crises Making me Collapse ? Measuring Transfer Learning Performances on Urgency Detection. In *19th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2022)*.

BREIMAN L. (2001). Random forests. *Machine learning*, **45**, 5–32.

CAILLAUT G., GRACIANNE C., ABADIE N., TOUYA G. & AUCLAIR S. (2022). Automated construction of a French Entity Linking dataset to geolocate social network posts in the context of natural disasters. In I. D. LIBRARY, Éd., *19th International Conference on Information Systems for Crisis Response and Management*, 19 th ISCRAM 2022 Conference Proceedings, Tarbes, France. HAL : hal-03631387.

CAMERON M. A., POWER R., ROBINSON B. & YIN J. (2012). Emergency situation awareness from twitter for crisis management. In *Proceedings of the 21st international conference on world wide web*, p. 695–698.

DE PRYCK K. (2022). *GIEC. La voix du climat*. Presses de Sciences Po.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186.

DIWERSY S., DIDIRKOVÁ I., DODANE C. & HIRSCH F. (2022). Les temps de la crise sanitaire au prisme d'une série chronologique : une étude phonético-textométrique. In *16th International Conference on Statistical Analysis of Textual Data*.

DUSART A., PINEL-SAUVAGNAT K. & HUBERT G. (2021). ISSumSet : a tweet summarization dataset hidden in a TREC track. In *Proceedings of the 36th annual ACM symposium on applied computing*, p. 665–671.

GIANNAKIDOU A. & MARI A. (2021). *Truth and veridicality in grammar and thought : Mood, modality, and propositional attitudes*. University of Chicago Press.

GRANO T. (2017). The logic of intention reports. *Journal of Semantics*, **34**(4), 587–632.

IMRAN M., CASTILLO C., LUCAS J., MEIER P. & VIEWEG S. (2014). Aidr : Artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW'2014, p. 159–162 : ACM. DOI : 10.1145/2567948.2577034.

IMRAN M., ELBASSUONI S., CASTILLO C., DIAZ F. & MEIER P. (2013). Extracting information nuggets from disaster-related messages in social media. *Iscram*, **201**(3), 791–801.

IMRAN M., MITRA P. & CASTILLO C. (2016). Twitter as a Lifeline : Human-annotated Twitter Corpora for NLP of Crisis-related Messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, (LREC'2016) : European Language Resources Association (ELRA).

INTERDONATO R., DOUCET A. & GUILLAUME J.-L. (2018). Unsupervised Crisis Information Extraction from Twitter Data. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, p. 579–580. DOI : 10.1109/ASONAM.2018.8508261.

KAYI E. S., NAN L., QU B., DIAB M. & MCKEOWN K. (2020). Detecting urgency status of crisis tweets : A transfer learning approach for low resource languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4693–4703.

KERSTEN J., KRUSPE A., WIEGMANN M. & KLAN F. (2019). Robust filtering of crisis-related tweets. In *ISCRAM 2019 conference proceedings-16th international conference on information systems for crisis response and management*.

KOZLOWSKI D., LANNELONGUE E., SAUDEMONT F., BENAMARA F., MARI A., MORICEAU V. & BOUMADANE A. (2020). A three-level classification of french tweets in ecological crises. *Information Processing & Management*, **57**(5), 102284.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2019). FlauBERT : Unsupervised Language Model Pre-training for French. *arXiv preprint arXiv :1912.05372*.

LIN Y., SHEN S., LIU Z., LUAN H. & SUN M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2124–2133, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/P16-1200.

MCCREADIE R., BUNTAIN C. & SOBOROFF I. (2019). TREC incident streams : Finding actionable information on social media. In *Proceedings of the 16th ISCRAM Conference*.

MCCREADIE R., BUNTAIN C. & SOBOROFF I. (2020). Incident streams 2019 : Actionable insights and how to find them. In *Proceedings of the 17th ISCRAM Conference*.

NGUYEN D., AL MANNAI K. A., JOTY S., SAJJAD H., IMRAN M. & MITRA P. (2017). Robust classification of crisis-related data on social networks using convolutional neural networks. In *Proceedings of the international AAAI conference on web and social media*, volume 11, p. 632–635.

NGUYEN D. T., MANNAI K. A. A., JOTY S., SAJJAD H., IMRAN M. & MITRA P. (2016). Rapid Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks. *arXiv preprint arXiv :1608.03902*.

OLTEANU A., VIEWEG S. & CASTILLO C. (2015). What to Expect When the Unexpected Happens : Social Media Communications Across Crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, p. 994–1009.

PALEN L. & LIU S. B. (2007). Citizen Communications in Crisis : Anticipating a Future of ICT-supported Public Participation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'07, p. 727–736.

REUTER C., HUGHES A. L. & KAUFHOLD M.-A. (2018). Social media in crisis management : An evaluation and analysis of crisis informatics research. *International Journal of Human–Computer Interaction*, **34**(4), 280–294.

SCHNEIDER K.-M. (2003). A comparison of event models for naive bayes anti-spam e-mail filtering. In *10th Conference of the European Chapter of the Association for Computational Linguistics*.

SEEBERGER P. & RIEDHAMMER K. (2022). Enhancing crisis-related tweet classification with entity-masked language modeling and multi-task learning. *arXiv preprint arXiv :2211.11468*.

TORAMAN C., KUCUKKAYA I. E., OZCELIK O. & SAHIN U. (2023). Tweets under the rubble : Detection of messages calling for help in earthquake disaster. *arXiv preprint arXiv :2302.13403*.

VIEWEG S., CASTILLO C. & IMRAN M. (2014). Integrating Social Media Communications into the Rapid Assessment of Sudden Onset Disasters. In *Proceedings of the 6th International Conference of Social Informatics*, SocInfo'14, p. 444–461.

WANG C., NULTY P. & LILLIS D. (2021). Transformer-based multi-task learning for disaster tweet categorisation. *arXiv preprint arXiv :2110.08010*.

ZAHERA H. M., ELGENDY I. A., JALOTA R., SHERIF M. A. & VOORHEES E. (2019). Fine-tuned bert model for multi-label tweets classification. In *TREC*, p. 1–7.

ZAHRA K., IMRAN M. & OSTERMANN F. O. (2020). Automatic identification of eyewitness messages on Twitter during disasters. *Information Processing & Management*, **57**(1), 102–107.

ZENG Q. & LI A.-R. (2022). A survey in automatic irony processing : Linguistic, cognitive, and multi-X perspectives. In *Proceedings of the 29th International Conference on Computational Linguistics*, p. 824–836, Gyeongju, Republic of Korea : International Committee on Computational Linguistics.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Outiller l'occitan : nouvelles ressources et lemmatisation

Aleksandra Miletić
Department of Digital Humanities, University of Helsinki
Yliopistonkatu 3, 00014 Helsinki, Finland
aleksandra.miletic@helsinki.fi

RÉSUMÉ

Ce travail présente des contributions récentes à l'effort de doter l'occitan de ressources et outils pour le TAL. Plusieurs ressources existantes ont été modifiées ou adaptées, notamment un tokéniseur à base de règles, un lexique morphosyntaxique et un corpus arboré. Ces ressources ont été utilisées pour entraîner et évaluer des modèles neuronaux pour la lemmatisation. Dans le cadre de ces expériences, un corpus plus large (2 millions de tokens) provenant du Wikipédia a été annoté en parties du discours et en lemmes. Le corpus est accessible sur Zenodo.

ABSTRACT

New Resources and Lemmatization Experiments for Occitan

This paper presents recent contributions to the creation of NLP tools and resources for Occitan. Several existing resources were modified or adapted, in particular a rule-based tokenizer, a morphosyntactic lexicon and a treebank. These resources were used to train and evaluate neural lemmatization models. As part of these experiments, a large corpus based on Wikipedia (2 million tokens) was POS-tagged and lemmatized. This new resource is shared through Zenodo.

MOTS-CLÉS : langues peu dotées, occitan, lemmatisation.

KEYWORDS: low-resourced languages, Occitan, lemmatization.

1 Introduction

L'occitan est une langue romane parlée majoritairement dans le sud de la France ainsi qu'en Italie et en Espagne. Il n'est pas standardisé et connaît une variation interne importante. L'occitan possède une riche littérature depuis les troubadours médiévaux, il est enseigné, de l'école à l'Université, mais n'a pas de statut officiel en France. Comme c'est souvent le cas des langues dans cette situation, les efforts pour construire des ressources et outils du TAL ont démarré tardivement par rapport à la majorité des langues officielles d'Europe. Néanmoins, à travers deux projets récents, RESTAURE [1] et LINGUATEC [2], les premiers corpus annotés (Bernhard *et al.*, 2018; Miletic *et al.*, 2020b) et ressources lexicales (Vergez-Couret, 2016; Bras *et al.*, 2020) ont été réalisés, ce qui a permis l'entraînement des premiers modèles d'étiquetage en parties du discours (Vergez-Couret & Urieli, 2015) et d'analyse syntaxique en dépendances (Miletic *et al.*, 2019b, 2020b). Notons cependant que ces expériences

1. RESTAURE : RESsources informatisées et Traitement AUtomatique pour les langues REgionales, convention ANR-14-CE24-0003-01. http://restaure.unistra.fr/

2. LINGUATEC : projet européen EFA 227/16 Développement de la coopération transfrontalière et du transfert de connaissances en technologies du langage. https://linguatec-poctefa.eu/fr/projet/

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 217–231.

s'appuient sur des méthodes d'apprentissage statistiques. Par ailleurs, à notre connaissance il n'y a pas encore de travaux publiés sur la lemmatisation de l'occitan.

La lemmatisation consiste à trouver la forme de base d'une forme fléchie donnée. La définition de la forme de base pour une catégorie grammaticale donnée peut varier d'une langue à l'autre. Elle peut consister, par exemple, à identifier le masculin singulier d'un adjectif (*bèlas* 'belles' > *bèu* 'beau') ou à retrouver l'infinitif d'une forme verbale (*calèva* 'fallait' > *caler* 'falloir'). Cette forme de traitement est particulièrement importante pour les langues à morphologie riche : elle permet de regrouper toutes les formes de surface provenant du même lemme et de diminuer ainsi la dispersion des données.

Les outils de lemmatisation basés sur l'apprentissage automatique statistique peuvent se diviser en deux groupes principaux : approches par arbres d'édition (*edit trees*) et approches par transduction de chaînes de caractères. La première méthode (Gesmundo & Samardzic, 2012; Grzegorz Chrupala & van Genabith, 2008; Müller *et al.*, 2015) consiste à identifier la série d'opérations d'édition qui permet d'obtenir le lemme à partir de la forme fléchie. Cette série d'opérations est attribuée à la paire *(forme fléchie, lemme)* en tant qu'étiquette. L'outil apprend l'arbre d'édition qui s'applique à chaque paire, ce qui permet d'aborder la lemmatisation comme une tâche de classification. Avec les méthodes neuronales, la lemmatisation a été redéfinie comme une tâche de transduction de chaînes de caractères (Bergmanis & Goldwater, 2018; Manjavacas *et al.*, 2019). L'avantage principal de ces méthodes par rapport à la génération précédente réside dans une meilleure capacité de généralisation aux tokens inconnus. Ce fait est particulièrement pertinent pour les langues à variation interne forte comme l'occitan, où on s'attend à un taux élevé de tokens inconnus.

Dans ce travail, nous décrivons des expériences en lemmatisation à l'aide d'outils basés sur des méthodes d'apprentissage neuronales. Nous détaillons également plusieurs modifications apportées aux outils et ressources existantes dans le cadre de ces expériences. Enfin, nous décrivons un corpus de 2 millions de tokens extrait du Wikipédia, que nous avons étiqueté, lemmatisé et diffusé.

2 Occitan : langue régionale peu dotée

L'occitan est une langue romane qui appartient au groupe gallo-roman, parlée dans une aire linguistique incluant 32 départements de la moitié sud de la France (à l'exception des zones catalane et basque), dans plusieurs vallées alpines d'Italie et au Val d'Aran en Espagne (cf. figure 2, empruntée à (Bernhard *et al.*, 2021)). La langue connaît une riche variation diatopique couplée à l'existence de plusieurs graphies. Aucune de ses variétés n'est reconnue comme standard, même si des standards dialectaux tendent à émerger. Ces facteurs contribuent à un important degré de variation de formes de surface, ce qui rend la création des ressources et outils pour le TAL plus exigeante que pour les langues standardisées.

2.1 Propriétés linguistiques et axes de variation

L'occitan est une langue *pro-drop* : la réalisation du sujet dans la phrase n'est pas obligatoire. Les formes verbales finies portent des marques de temps, de personne et de nombre. Dans de nombreux dialectes, le nombre et le genre sont marqués sur tous les éléments du groupe nominal. À la différence du français, l'occitan préserve l'utilisation du passé simple et du subjonctif de l'imparfait, y compris à l'oral. L'exemple en provençal donné en figure 1 illustre certaines de ces propriétés.

(1) a. Totei lei personas naisson liuras e egalas en dignitat e en drech.
b. Tóuti li persouno naisson liéuro e egalo en dignita e en dre.
‘Toutes les personnes naissent libres et égales en dignité et en droit.’

FIGURE 1 – Première phrase de la Déclaration des droits de l’homme en provençal, en graphie classique (a) et en graphie mistralienne (b)

La classification dialectale la plus répandue (Bec, 1995) regroupe les variétés de l’occitan en six principaux dialectes : l’auvergnat, le gascon, le languedocien, le limousin, le provençal et le vivaro-alpin (v. figure 2) . Ces dialectes connaissent à leur tour une variation interne plus ou moins prononcée.

FIGURE 2 – Carte de dialectes

La variation diatopique est présente à tous les niveaux de fonctionnement linguistique. Au niveau lexical, on trouve des cas de figure comme celui du mot *pomme de terre* : en gascon, on utilise la forme *mandòrra*, alors qu’en languedocien on utilise plutôt *trufa/trufet* ou *patana/patanon*. Différents dialectes ayant subi des processus phonétiques différents, on retrouve des séries de formes spécifiques à chaque dialecte, à l’instar du mot *fils*, qui correspond à *hilh* en gascon, à *filh* en languedocien et en limousin, et à *fiu* en provençal. Au niveau morphosyntaxique, la flexion verbale varie au niveau inter-dialectal, mais aussi intra-dialectal. À partir du paradigme flexionnel le plus répandu dans chaque dialecte, la forme de la première personne du pluriel du présent du verbe *être* correspond à *èm* en gascon, *sem* en limousin, *sèm* ou *sièm* en languedocien et *siam* en provençal. La variation syntaxique la plus prononcée est celle du gascon par rapport aux autres dialectes. À titre d’illustration, les clitiques objet et les pronoms réflexifs se trouvent en position post-verbale plus souvent qu’en languedocien, où ils sont typiquement préverbaux (cf. gasc. *ne cau pas està**’s** darrèr mieidia* vs lang. ***vos*** *cal pas demorar après miègjorn*, ‘vous ne devriez pas rester après midi’). Le gascon, à la différence de tous les autres dialectes de l’occitan, exhibe également des particules énonciatives, qui marquent la modalité de la phrase. Ces particules occupent la position entre le sujet (s’il est réalisé) et le verbe, et elles sont obligatoires dans certains parlers gascons. Comparez, *e.g.*, gasc. *Lo vent* ***que*** *s’èra lhevat et* ***que*** *hasó drin fresc vs* lang. *Lo vent s’èra levat et fasiá un pauc freg* ‘Le vent s’était levé et il faisait un peu froid’.

En outre, plusieurs normes d’écriture existent, dont deux sont dominantes aujourd’hui : la norme dite *mistralienne*, proche des conventions orthographiques françaises, et la norme dite *classique*, fondée sur l’orthographe utilisée par les troubadours occitans (Sibille, 2002). L’exemple 1 illustre la

différence entre ces deux normes en reprenant la même phrase dans le même dialecte (le provençal) en graphie classique (1a) et en graphie mistralienne (1b). À ces différences s'ajoute la variation au niveau individuel : il n'est pas rare qu'un rédacteur adopte une graphie qui lui est propre.

2.2 Situation de l'occitan en TAL

Bien que l'occitan ait été reconnu comme appartenant à l'héritage culturel de la France [3], il n'est pas reconnu comme langue officielle dans ce pays, où réside la majorité de ses locuteurs. Comme il est souvent le cas dans cette situation, l'outillage de l'occitan a démarré plus tardivement par rapport à celui de la majorité de langues officielles européennes. Deux projets majeurs ont permis de constituer, par exemple, un corpus étiqueté en parties du discours (Bernhard *et al.*, 2018), un lexique (Vergez-Couret, 2016; Bras *et al.*, 2020) et un corpus arboré (Miletic *et al.*, 2020a,b). Des expériences en étiquetage en parties du discours (Vergez-Couret & Urieli, 2015) et en analyse syntaxique automatique (Miletic *et al.*, 2019b, 2020b) ont donné lieu aux premiers modèles de traitement automatique de l'occitan. Un modèle de synthèse de la parole a été développé récemment (Corral *et al.*, 2020).

En ce qui concerne les corpus larges, l'occitan dispose d'une base de données textuelles intérrogeable en ligne (Bras & Vergez-Couret, 2016). Un corpus de 2 millions de tokens extrait des articles du Wikipédia en occitan est diffusé au sein de *Leipzig Corpora Collection* (Goldhahn *et al.*, 2012). Le corpus multilingue OSCAR, basé sur du contenu moissonné à travers CommonCrawl [4], comprend également un sous-corpus occitan (Ortiz Suárez *et al.*, 2019). Enfin, un corpus basé sur des discussions Wikipédia a également été publié (Miletić & Scherrer, 2022a). Ce travail a été accompagnée d'expériences en identification de l'occitan dans un contenu multilingue (Miletić & Scherrer, 2022b).

Quant aux modèles de langue pré-entraînés, des plongements lexicaux pour l'occitan dérivés en utilisant fastText existent (Bojanowski *et al.*, 2017; Costa-jussà *et al.*, 2022), et l'occitan est également représenté dans le modèle contextualisé multilingue mBERT (Devlin *et al.*, 2019).

Notons que les expériences en étiquetage en parties du discours et en analyse syntaxique automatique citées ci-dessus ont été réalisés à l'aide des methodes d'apprentissage statistiques. À notre connaissance, en dehors d'expériences non encore publiées dans le cadre d'une thèse en cours sur la variation de l'occitan à l'épreuve des outils de TAL (Poujade, en préparation), il n'existe pas à ce jour de travaux sur l'occitan qui évalueraient l'efficacité des méthodes neuronales pour le traitement automatique de cette langue. Par ailleurs, nous n'avons pas identifié de travaux sur la lemmatisation de l'occitan. Or, ce niveau de traitement est crucial pour les langues à variation importante : comme la lemmatisation consiste à fournir pour chaque forme fléchie la forme de base correspondante, elle permet de réduire la dispersion de données dans un corpus et facilite ainsi l'apprentissage automatique.

3 Modification de ressources existantes

3.1 Tokénisation

Un tokéniseur à base de règles, implémenté en Perl, a été développé dans le cadre du projet RESTAURE pour le gascon et le languedocien (Vergez-Couret, 2019). Il met en place un traitement

3. Cf. Article 75-1 de la Constitution de la Cinquième République française.
4. https://commoncrawl.org/

spécial basé sur des listes d'unités polylexicales : elles ne sont pas tokénisées en formes graphiques individuelles, mais sont plutôt traitées en token unique (cf. *tre que* 'dès que' → [*tre_que*]).

La nouvelle version du tokéniseur est réalisée en Python. Nous reprenons les règles de base telles que définies dans la documentation de l'outil, mais abandonnons le traitement spécial des unités polylexicales. En effet, nous visons à produire une tokénisation plus compatible avec les exigences du projet Universal Dependencies [5], auxquelles se conforme également le corpus arboré Tolosa Treebank (Miletic *et al.*, 2020b). Comme le projet Universal Dependencies ne permet pas le maintien de ce type de tokens, nous effectuons une tokénisation en formes graphiques simples.

L'un des points principaux à traiter concerne le statut de l'apostrophe. À la différence du français, où l'apostrophe appartient systématiquement au token de gauche (cf. *l'apostrophe* → [*l', apostrophe*]), en occitan il peut appartenir aussi bien au token de gauche qu'à celui de droite. Par exemple, l'article défini se comporte d'une manière comparable au français : *l'acadèmia* → [*l', acadèmia*]. Le traitement diffère pour certains clitiques en post-position : *Ne'm vòs pas ?* 'Tu ne me veux pas ?' doit être tokénisée en [*Ne, 'm, vòs, pas, ?*]. Les différents cas de figure sont traités par des règles de tokénisation correspondantes, définies lors de la création du tokéniseur originel.

3.2 Conversion du lexique Loflòc vers le format UD

Loflòc (Lexic obèrt flechit occitan - Lexique ouvert fléchi occitan) (Vergez-Couret, 2016; Bras *et al.*, 2020) est un lexique morphosyntaxique développé dans le cadre du projet ANR RESTAURE puis du projet européen POCTEFA LINGUATEC, en collaboration avec Lo Congrès Permanent de la Lenga Occitana [6]. La version utilisée ici contient 849 605 entrées correspondant à 58 373 lemmes ; le contenu est en languedocien. Les informations morphosyntaxiques sont encodées à l'aide du jeu d'étiquettes GRACE (Rajman *et al.*, 1997). Il s'agit d'étiquettes positionnelles, où la première position encode la catégorie grammaticale, la deuxième la sous-catégorie sémantique, et les positions restantes les traits morphosyntaxiques pertinents pour la catégorie en question. Ainsi, l'étiquette `Ncfs` représente un nom (`N`) commun (`c`) féminin (`f`) singulier (`s`). La définition du jeu d'étiquettes pour l'occitan est disponible dans le manuel d'annotation du corpus étiqueté développé dans le cadre du projet RESTAURE [7].

Couplée à un corpus d'entraînement, cette ressource a le potentiel de faciliter la création de modèles performants pour le traitement automatique de l'occitan. Or, le corpus arboré Tolosa Treebank suit le schéma d'annotation du projet Universal Dependencies. Pour éliminer cette incompatibilité, nous avons effectué une conversion du lexique Loflòc vers le format UD. Le format UD encode les informations morphosyntaxiques à deux niveaux : les parties du discours sont exprimées en utilisant un jeu d'étiquettes réduit de 17 étiquettes [8], alors que les traits morphosyntaxiques sont encodés à l'aide de paires `Trait=Valeur` [9]. Notre script de conversion répartit les informations encodées dans les étiquettes GRACE entre ces deux niveaux. Une conversion GRACE → UD a été appliquée au premier corpus occitan étiqueté en parties du discours (Miletic *et al.*, 2019a), mais elle se limitait au niveau des étiquettes POS. Des exemples de conversion sont donnés dans le tableau 1 et la distribution d'entrées par partie du discours après la conversion est présentée dans le tableau 2.

5. https://universaldependencies.org/docs/u/overview/tokenization.html
6. https://locongres.org/fr/
7. https://doi.org/10.5281/zenodo.1173113
8. https://universaldependencies.org/u/pos/all.html
9. https://universaldependencies.org/u/feat/index.html

Forme	Lemme	GRACE	UD UPOS	UD Traits morphosyntaxiques
abausada	abausat	Afpfs	ADJ	Gender=Fem|Number=Sing|Degree=Pos
abscèsses	abscès	Ncmp	NOUN	Gender=Masc|Number=Plur
agirai	agir	Vmi-f1s-	VERB	Mood=Ind|Tense=Fut|Person=1|Number=Sing

TABLE 1 – Exemple de transformation GRACE → UD

Etiquette	Catégorie	Occurr.	Etiquette	Catégorie	Occurr.
ADJ	adjectif	58 995	NUM	numéral	231
ADP	adposition	670	PRON	pronom	519
ADV	adverbe	1 841	PROPN	nom propre	1 839
DET	déterminant	363	VERB	verbe	690 025
NOUN	nom commun	94 073			

TABLE 2 – Distribution d'entrées par partie du discours dans le lexique Loflòc au format UD

3.3 Ré-échantillonnage et correction de Tolosa Treebank

Le corpus arboré Tolosa Treebank contient 26 mille tokens annotés en parties du discours, en lemmes et en dépendances syntaxiques suivant le schéma d'annotation Universal Dependencies. Le languedocien était le premier dialecte à être annoté et il reste majoritaire dans le corpus, mais des textes en gascon, limousin et provençal ont également été intégrés. Tous les textes sont en graphie classique. La décision d'initalement limiter le corpus à un dialecte et à une graphie a été motivée par le besoin de pouvoir contrôler les différents niveaux de variation (Miletic *et al.*, 2020a). Le choix du languedocien est dû à sa position centrale dans le continuum dialectal : on peut s'attendre à ce que le contenu annoté en languedocien soit le plus utile pour le traitement des autres dialectes. La stratification par dialecte dans la version actuelle du corpus est donnée dans le tableau 3.

	Phrases	Tokens	Types	L. phrase	T = L
Tous dialectes	1 522	26 122	6 196	17,16	64,7
Gascon	255	4 170	1 429	16,35	64,5
Languedocien	1 113	19 315	4 499	17,35	65,1
Limousin	77	1 344	596	17,45	63,8
Provençal	77	1 293	583	16,79	61,4

TABLE 3 – Contenu de Tolosa Treebank par dialecte. L. phrase : longueur moyenne de la phrase en tokens. T = L : % de tokens identiques à leur lemme.

La première version de ce corpus à quatre dialectes a été utilisée pour des expériences en analyse syntaxique automatique trans-dialectale (Miletic *et al.*, 2020b). Dans ce but, le corpus avait été divisé en échantillons d'entraînement et de test, mais un échantillon de développement (*dev set*) n'a pas été créé car les méthodes utilisées dans ces expériences ne l'exigeaient pas. Comme il est en général requis par les méthodes neuronales pour l'évaluation des modèles intermédiaires durant l'entraînement, nous opérons un nouveau découpage du corpus en échantillons *train*, *test* et *dev* pour le languedocien et le gascon. Pour le limousin et le provençal, la quantité de données actuellement annotées ne permet pas la création d'un échantillon supplémentaire. Ce redécoupage du corpus

	train			test					dev				
	Phrases	Tokens	Types	Phrases	Tokens	Types	Inc.	Amb.	Phrases	Tokens	Types	Inc.	Amb.
Tous dialectes	1 196	20 551	5 292	202	3 179	1 054	22,11	28,18	124	2 392	1 009	16,39	31,77
Gascon	195	3 258	1 173	35	421	230	26,37	23,28	25	491	267	19,35	33,60
Languedocien	884	15 494	3 937	130	1 920	577	19,64	27,50	99	1 901	814	15,62	31,30
Limousin	56	919	434	16	413	211	27,76	31,76	-	-	-	-	-
Provençal	61	880	424	16	413	211	23,49	32,69	-	-	-	-	-

TABLE 4 – Division de Tolosa Treebank en échantillons d'entraînement, de développement et de test. Inc : % de tokens inconnus. Amb : % de tokens ambigus.

signifie que les résultats obtenus sur les versions différentes ne seront pas directement comparables.

Des informations quantitatives sur les différents échantillons, pour l'ensemble du corpus et par dialecte, sont proposées dans le tableau 4. Pour les échantillons *dev* et *test*, nous ajoutons le pourcentage de tokens inconnus et ambigus par rapport à l'échantillon *train* correspondant. Un token est considéré comme inconnu s'il est absent de l'échantillon *train*, alors qu'un token ambigu y figure, mais se trouve associé à plusieurs annotations possibles. Comme nos expériences se focalisent sur la lemmatisation, le pourcentage de tokens ambigus correspond à la part de tokens qui sont associés à plusieurs lemmes.

4 Expériences en lemmatisation

L'objectif des expériences présentées ici est d'identifier des stratégies utiles pour la lemmatisation de l'occitan. Les stratégies explorées concernent trois dimensions principales : le paradigme d'apprentissage (adaptation de l'entraînement séquentiel *vs* l'entraînement joint appliqués à l'étiquetage et à la lemmatisation), la taille et la nature de données d'entraînement (utilité d'un corpus limité annoté manuellement *vs* un corpus large annoté automatiquement) et la gestion de la variation dialectale (efficacité de modèles spécifiques à chaque dialecte *vs* un modèle global). Nous entraînons également des modèles d'étiquetage en POS afin d'évaluer les outils de lemmatisation sur des étiquettes POS fournies automatiquement. Les expériences discutées ici ont été mises en parallèle avec des expériences sur la lemmatisation du bas saxon (Miletić & Siewert, 2023), permettant d'identifier des difficultés et des stratégies communes, particulièrement pertinentes dans le contexte des langues non standardisées et peu dotées.

4.1 Outils utilisés

MaChAmp (van der Goot *et al.*, 2021) : cet outil permet d'effectuer l'apprentissage multi-tâche et le paramétrage (*fine-tuning*) d'un éventail de tâches du TAL, y compris l'étiquetage en parties du discours, la lemmatisation, l'analyse syntaxique automatique, la modélisation du langage masquée et la génération de texte. MaChAmp utilise un modèle contextualisé pré-entraîné comme encodeur et effectue le *fine-tuning* en fonction des tâches en aval qui lui sont demandées. Chaque tâche dispose d'un décodeur dédié qui permet d'effectuer les prédictions pour la tâche en question. L'outil permet également de réaliser un entraînement initial pour une tâche donnée, puis de ré-entraîner le modèle pour la même tâche sur un deuxième jeu de données. Nous avons exploité cette fonctionnalité dans nos expériences de lemmatisation. Par défaut, MaChAmp utilise les plongements lexicaux du modèle

pré-entraîné mBERT (Devlin *et al.*, 2019).

Stanza NLP (Qi *et al.*, 2020) : cette chaîne de traitement intègre actuellement de modèles de traitement pour 66 langues différentes (mais pas pour l'occitan). Elle comprend des modules de tokénisation, d'analyse des tokens multi-mots, de lemmatisation, d'étiquetage en parties du discours et en traits morphosyntaxiques, d'analyse syntaxique en dépendances et de reconnaissance des entités nommées. Comme Stanza permet l'entraînement de nouveaux modèles, nous avons utilisé son étiqueteur, basé sur un modèle biLSTM, et son lemmatiseur, qui intègre un modèle neuronal de séquence à séquence (*seq2seq*). L'étiqueteur permet d'utiliser des plongements lexicaux statiques ; nous avons tiré profit de cette fonctionnalité en utilisant ceux de fastText [10].

4.2 Préannotation d'un corpus plus large

Le corpus Tolosa Treebank est d'une taille relativement restreinte (26K tokens). Nous avons donc souhaité évaluer l'utilité d'un corpus plus large, qui serait annoté de manière automatique. Pour ce faire, nous nous sommes servie du corpus extrait du Wikipédia en occitan diffusé dans *Leipzig Corpora Collection* (Goldhahn *et al.*, 2012). La version téléchargeable de ce corpus contient 100K phrases correspondant à 2M de tokens [11]. Le contenu est segmenté en phrases, mais pas annoté.

Afin d'effectuer l'étiquetage en parties du discours et la lemmatisation de ce corpus, nous avons utilisé l'outil MaChAmp. Dans ces expériences initiales avec l'outil, nous avons opté pour l'entraînement indépendant des modèles d'étiquetage et de lemmatisation. Les modèles ont été appris sur l'échantillon d'entraînement du corpus Tolosa Treebank. Nous avons utilisé les plongements lexicaux par défaut.

L'étiqueteur en parties du discours a atteint l'exactitude de 92,26 % sur l'échantillon de test contenant les quatre dialectes. Le résultat le plus élevé a été obtenu sur le languedocien (92,97 %) et le plus bas sur le provençal (89,10 %). L'exactitude globale du lemmatiseur a atteint 89,30 %, le résultat le plus élevé étant 93,33 % sur le languedocien et le limousin, et le plus bas 88,6 % sur le gascon.

Pour annoter le corpus Wikipédia, nous l'avons d'abord tokénisé à l'aide du tokéniseur présenté dans la section 3.1. Si un token disposait d'une entrée non ambiguë dans le lexique Loflòc, nous avons retenu l'information disponible dans le lexique. Dans le cas contraire, nous avons fait appel aux modèles MaChAmp. Environ un tiers des tokens ont été annotés à partir du lexique.

4.3 Résultats et discussion

Cette section est dédiée à la discussion des résultats de nos expériences en lemmatisation. Nous indiquons systématiquement l'exactitude moyenne et la déviation standard obtenues à partir de trois tours d'entraînement avec des *random seeds* différents. Nous évaluons les outils sur l'ensemble des tokens, mais aussi sur les tokens inconnus et ambigus. Les résultats discutés ici ont été obtenus sur les échantillons *test*. Les résultats sur les échantillons *dev* sont disponibles dans l'annexe A.

Nous avons d'abord évalué les performances de plusieurs modèles globaux, entraînés et évalués sur le contenu en quatre dialectes. Les résultats sont présentés dans le tableau 5. La colonne *train* indique le corpus d'entraînement : TTB correspond à Tolosa Treebank, WIKI au corpus Wikipédia et COMB à la combinaison des deux. Dans le cas de MaChAmp, WIKI+TTB indique un premier entraînement sur

10. https://fasttext.cc/docs/en/crawl-vectors.html
11. https://corpora.uni-leipzig.de/en?corpusId=oci_wikipedia_2021

le corpus Wikipédia suivi d'un ré-entraînement sur le Tolosa Treebank. La colonne *tâche* précise si le modèle a été entraîné sur la lemmatisation seule ou sur l'étiquetage aussi. *Cond. entraîn.* et *Cond. test* indiquent respectivement les informations utilisées pour l'entraînement et pour l'évaluation. Les modèles qui s'appuient sur les étiquettes POS ont été évalués en utilisant l'annotation automatique produite par le modèle d'étiquetage entraîné sur le même corpus.

Dans ce scénario, les meilleurs résultats ont été obtenus par le modèle Stanza entraîné sur le Tolosa Treebank en s'appuyant sur les étiquettes POS (exactitude globale : 93,21 %). Quant à MaChAmp, le modèle le plus performant est celui entraîné sur le corpus Wikipédia et ré-entrâiné sur le Tolosa Treebank dans le paradigme d'apprentissage joint (exactitude globale : 92,16 %). Ces modèles dépassent ceux entraînés sans accès à l'étiquetage en POS, ce qui confirme encore une fois l'utilité des étiquettes POS pour la lemmatisation.

Globalement, les résultats de MaChAmp varient relativement peu : la différence entre les modèles le moins et le plus performant est de <1 %. Le passage du corpus plus petit au corpus plus large annoté automatiquement, ainsi que le ré-entraînement de ce deuxième modèle sur le corpus *gold* apportent tous les deux des améliorations, mais celles-ci restent limitées (<0,5 %). Quant à Stanza, les différences entre les modèles sont plus prononcées, les scores allant de 90,35 % pour le modèle entraîné sur le Tolosa Treebank sans utiliser les étiquettes POS à 93,21 % pour le modèle entraîné sur le même corpus en exploitant l'annotation morphosyntaxique. Il est intéressant que l'ajout du corpus Wikipédia au corpus Tolosa Treebank ici n'apporte pas d'amélioration (92,49 % vs 93,21 %). Il semblerait donc que pour MaChAmp la quantité de données d'entraînement est plus importante que la qualité de l'annotation, alors que pour Stanza ce serait l'inverse.

La question de la fiabilité de l'annotation semble particulièrement importante pour le traitement des tokens inconnus et ambigus : pour les deux outils, les meilleurs scores sur ces deux catégories ont été obtenus par les modèles entraînés sur le corpus *gold*. L'utilisation du corpus annoté automatiquement entraîne des pertes d'environ 4-5 % sur les tokens inconnus et d'environ 3-4 % sur les tokens ambigus pour MaChAmp, alors que pour Stanza les pertes sont respectivement d'environ 10 % et 4 %.

Dans un deuxième temps, nous avons évalué l'adaptation de différents modèles à chacun des dialectes (cf. tableau 9). Nous avons également développé des modèles MaChAmp ciblés en effectuant le ré-entraînement sur chacun des sous-corpus (cf. les modèles WIKI+GA, WIKI+LI, WIKI+LA, WIKI+PR) pour identifier la meilleure stratégie pour les dialectes individuels.

Le modèle Stanza entraîné sur le Tolosa Treebank reste le plus utile : il atteint les meilleurs résultats sur les trois catégories de tokens pour le gascon, le languedocien et le provençal, et il est également le plus performant sur les tokens ambigus sur l'échantillon limousin. Pour ce dernier dialecte, c'est le modèle MaChAmp ré-entraîné sur l'ensemble du corpus Tolosa Treebank qui gagne sur les tokens inconnus et sur l'ensemble des tokens. Pour les quatre dialectes, le ré-entraînement sur les sous-corpus dédiés donne des résultats moins élevés que l'utilisation de l'ensemble du corpus Tolosa Treebank. Les pertes sont les plus prononcées sur le limousin et le provençal (respectivement d'environ 3 % et 5 %). Comme ces deux dialectes disposent des sous-corpus les plus petits, il est possible que cet effet soit dû à la taille de l'échantillon de ré-entraînement.

Globalement, l'utilité d'un corpus large annoté automatiquement semble dépendre de l'outil : avec MaChAmp, le corpus plus large apporte une amélioration au niveau du modèle global, mais ce n'est pas le cas de Stanza. Cet outil semble favoriser la fiabilité de l'annotation et obtient les meilleurs résultats à partir du corpus *gold*. Plus généralement, sur les token inconnus, les modèles entrâinés sur le corpus *gold* surpassent les modèles entraînés sur le corpus plus large. Si l'on vise à optimiser

Outil	*train*	Tâche	Cond. entraîn.	Cond. test	Tous	Inconnus	Ambigus
MaChAmp	TTB	LEM	no POS, gold LEM	no POS	$91,28^{\pm 0,42}$	$72,22^{\pm 1,55}$	$96,23^{\pm 0,37}$
	WIKI	POS+LEM	pred. POS+LEM	no POS	$91,77^{\pm 0,23}$	$68,54^{\pm 1,86}$	$92,19^{\pm 0,14}$
	WIKI+TTB	POS+LEM	pred. POS+LEM	no POS	$92,16^{\pm 0,25}$	$67,20^{\pm 0,33}$	$93,05^{\pm 0,45}$
Stanza	TTB	LEM	no POS, gold LEM	no POS	$90,35^{\pm 0,42}$	$66,86^{\pm 1,85}$	$95,78^{\pm 0,00}$
	TTB	LEM	gold POS+LEM	pred. POS	**93,21** $^{\pm 0,09}$	**78,43** $^{\pm 0,41}$	**96,69** $^{\pm 0,00}$
	COMB	LEM	pred. POS+LEM	pred. POS	$92,49^{\pm 0,08}$	$68,40^{\pm 0,98}$	$92,63^{\pm 0,00}$

TABLE 5 – Lemmatisation : résultats sur l'ensemble de l'échantillon de test

		Gascon				**Limousin**		
Outil	*train*	Tous	Inconnus	Ambigus	*train*	Tous	Inconnus	Ambigus
MaChAmp	WIKI+TTB	$89,66^{\pm 0,52}$	$57,01^{\pm 1,24}$	$90,28^{\pm 0,57}$	WIKI+TTB	**90,91** $^{\pm 0,20}$	**74,42** $^{\pm 1,90}$	$94,35^{\pm 0,46}$
	WIKI+GA	$88,86^{\pm 0,41}$	$54,38^{\pm 1,24}$	$89,58^{\pm 0,98}$	WIKI+LI	$87,64^{\pm 0,57}$	$64,34^{\pm 1,10}$	$92,66^{\pm 0,80}$
Stanza	TTB	**90,71** $^{\pm 0,75}$	**77,78** $^{\pm 2,79}$	**91,49** $^{\pm 0,00}$	TTB	$90,59^{\pm 0,41}$	$72,60^{\pm 0,80}$	**99,22** $^{\pm 0,00}$
	COMB	$90,06^{\pm 0,11}$	$67,54^{\pm 1,24}$	$89,58^{\pm 0,00}$	COMB	$89,79^{\pm 0,23}$	$66,67^{\pm 1,09}$	$92,66^{\pm 0,00}$
		Languedocien				**Provençal**		
Outil	*train*	Tous	Inconnus	Ambigus	*train*	Tous	Inconnus	Ambigus
MaChAmp	WIKI+TTB	$93,08^{\pm 0,48}$	$69,91^{\pm 0,33}$	$92,76^{\pm 0,69}$	WIKI+TTB	$91,67^{\pm 0,00}$	$54,67^{\pm 1,89}$	$95,14^{\pm 0,44}$
	WIKI+LA	$92,56^{\pm 0,60}$	$68,29^{\pm 0,33}$	$92,29^{\pm 0,80}$	WIKI+PR	$86,60^{\pm 0,11}$	$52,00^{\pm 0,00}$	$89,55^{\pm 0,25}$
Stanza	TTB	**94,42** $^{\pm 0,13}$	**81,35** $^{\pm 0,90}$	**96,54** $^{\pm 0,00}$	TTB	**92,81** $^{\pm 0,31}$	**74,92** $^{\pm 1,28}$	**98,51** $^{\pm 0,00}$
	COMB	$93,72^{\pm 0,11}$	$71,53^{\pm 1,50}$	$92,98^{\pm 0,00}$	COMB	$92,08^{\pm 0,12}$	$54,67^{\pm 1,89}$	$93,51^{\pm 0,00}$

TABLE 6 – Lemmatisation : résultats par dialecte

les performances des modèles sur cette catégorie de tokens, il semble utile de favoriser la qualité de l'annotation plutôt que la taille du corpus d'entraînement. Enfin, avec MaChAmp, les entraînements ciblés par dialecte n'apportent pas d'amélioration par rapport à l'entraînement global. Néanmoins, la taille de nos sous-corpus étant encore limitée, notamment pour le limousin et le provençal, cette stratégie mériterait d'être ré-évaluée sur des versions futures du corpus.

5 Corpus Wikipédia annoté

Afin de favoriser les travaux sur l'occitan, nous distribuons le corpus Wikipédia étiqueté en parties du discours et lemmatisé dans le cadre de nos expériences. Les 2 millions de tokens correspondent à 111 656 lemmes différents. La distribution des parties du discours se trouve dans le tableau 7. Dans la version actuelle du corpus, nous gardons l'annotation initiale (voir section 4.2). Celle-ci évoluera à l'avenir afin de prendre en compte les résultats présentés ici. Le corpus est disponible à l'adresse suivante : https://doi.org/10.5281/zenodo.7777340.

6 Conclusion

Nous avons présenté plusieurs avancées dans les efforts pour outiller l'occitan, une langue régionale non standardisée. Une part de ces efforts était focalisée sur des ressources existantes : des améliorations mineures ont été apportées à un tokéniseur existant et au corpus arboré Tolosa Treebank, alors

Etiquette	Catégorie	Occ.	Etiquette	Catégorie	Occ.
ADJ	adjectif	135 760	NUM	numéral	45 116
ADP	adposition	306 798	PART	particule	4 610
ADV	adverbe	76 063	PRON	pronom	65 916
AUX	verbe auxiliaire	55 023	PROPN	nom propre	92 554
CCONJ	conj. de coord.	56 173	PUNCT	ponctuation	225 596
DET	déterminant	317 677	SCONJ	subordonnant	20 831
INTJ	interjection	3 589	VERB	verbe	191 342
NOUN	nom commun	434 306	X	foreign	1 654

TABLE 7 – Distribution de parties du discours dans le corpus Wikipédia

que le lexique Loflòc a été converti du format GRACE vers le format Universal Dependencies. Deuxièmement, nous avons présenté, à notre connaissance, la première évaluation en lemmatisation de l'occitan. Ces expériences ont été réalisées à l'aide d'outils neuronaux, qui ne figurent pas pour l'heure dans les travaux publiés sur le traitement de l'occitan. Nous avons également décrit une nouvelle ressource : un corpus extrait du Wikipédia, qui a été tokénisé, étiqueté en parties du discours et lemmatisé. Ce corpus est désormais disponible sur Zenodo.

Les modèles de lemmatisation que nous avons entraînés atteignent des résultats solides, avec un taux d'exactitude allant de 90,71 % sur le gascon jusqu'à 94,42 % sur le languedocien, l'exactitude globale sur l'ensemble des dialectes étant de 93,21 %. Comme nous avons entraîné tous les modèles avec les valeurs des paramètres par défaut, il est possible que des tests de paramétrage apportent des gains supplémentaires. L'utilisation d'un corpus large annoté automatiquement a amélioré les résultats avec l'outil MaChAmp ; en revanche, Stanza s'est montré plus performant en limitant l'entraînement aux données annotées manuellement. Quant aux entraînements ciblés pour chaque dialecte, nous n'avons pas observé d'effets positifs ; qui plus est, cette stratégie entraîne des pertes importantes pour le limousin et le provençal. Ces observations sont sans doute liées à la taille limitée des sous-corpus individuels ; elles méritent d'être ré-évaluées sur les futures versions du corpus Tolosa Treebank.

Nous espérons que ces résultats, ainsi que les ressources présentées ici, favoriseront la poursuite des travaux sur l'occitan et faciliteront sa sortie du statut de langue peu dotée.

Remerciements

Je souhaite remercier Myriam Bras (Université de Toulouse) et Marianne Vergez-Couret (Université de Poitiers) pour leurs retours précieux.

Ce travail a été effectué dans le cadre du projet "CorCoDial – Corpus-based computational dialectology" (Academy of Finland, No. 342859).

A Résultats sur l'échantillon *dev*

Outil	*train*	Tâche	Cond. entraîn.	Cond. test	Tous	Inconnus	Ambigus
MaChAmp	TTB	LEM	no POS, gold LEM	no POS	$93,57^{\pm 0,06}$	$78,74^{\pm 1,14}$	$95,08^{\pm 0,28}$
	WIKI	POS+LEM	pred. POS+LEM	no POS	$93,32^{\pm 0,09}$	$76,07^{\pm 0,50}$	$91,94^{\pm 0,15}$
	WIKI+TTB	POS+LEM	pred. POS+LEM	no POS	$94,24^{\pm 0,17}$	$73,49^{\pm 0,74}$	$93,47^{\pm 0,29}$
Stanza	TTB	LEM	no POS, gold LEM	no POS	$92,84^{\pm 0,14}$	$75,43^{\pm 0,84}$	$93,10^{\pm 0,0}$
	TTB	LEM	gold POS+LEM	pred. POS	$94,68^{\pm 0,03}$	$83,16^{\pm 0,21}$	$94,86^{\pm 0,0}$
	COMB	LEM	pred. POS+LEM	pred. POS	$93,53^{\pm 0,06}$	$74,01^{\pm 1,11}$	$91,19^{\pm 0,0}$

TABLE 8 – Exactitude de lemmatisation sur tous les dialectes. Échantillon *dev*.

		Gascon				**Languedocien**		
Outil	*train*	Tous	Inconnus	Ambigus	*train*	Tous	Inconnus	Ambigus
MaChAmp	WIKI+TTB	$93,60^{\pm 0,36}$	$69,10^{\pm 1,15}$	$93,55^{\pm 1,11}$	WIKI+S	$94,40^{\pm 0,14}$	$75,58^{\pm 0,95}$	$93,45^{\pm 0,1}$
	WIKI+GA	$92,83^{\pm 0,10}$	$69,10^{\pm 2,30}$	$92,14^{\pm 0,22}$	WIKI+LA	$94,12^{\pm 0,13}$	$74,03^{\pm 1,09}$	$93,25^{\pm 0,11}$
Stanza	WIKI	$94,29^{\pm 0,20}$	$79,65^{\pm 0,99}$	$96,15^{\pm 0,00}$	TTB	$94,78^{\pm 0,02}$	$84,29^{\pm 0,16}$	$94,51^{\pm 0,0}$
	COMB	$90,40^{\pm 0,00}$	$68,29^{\pm 0,00}$	$87,26^{\pm 0,00}$	COMB	$94,33^{\pm 0,08}$	$76,75^{\pm 1,65}$	$92,16^{\pm 0,0}$

TABLE 9 – Exactitude de lemmatisation par dialecte. Échantillon *dev* (le corpus utilisé propose des échantillons *dev* seulement pour le gascon et le languedocien.)

Références

BEC P. (1995). *La langue occitane*. PUF, 6th édition.

BERGMANIS T. & GOLDWATER S. (2018). Context sensitive neural lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1391–1400, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1126.

BERNHARD D., LIGOZAT A.-L., BRAS M., MARTIN F., VERGEZ-COURET M., ERHART P., SIBILLE J., TODIRASCU A., BOULA DE MAREÜIL P. & HUCK D. (2021). Collecting and annotating corpora for three under-resourced languages of france : Methodological issues. *Language Documentation & Conservation*, (15), 316–357.

BERNHARD D., LIGOZAT A.-L., MARTIN F., BRAS M., MAGISTRY P., VERGEZ-COURET M., STEIBLÉ L., ERHART P., HATHOUT N., HUCK D., REY C., REYNÉS P., ROSSET S., SIBILLE J. & LAVERGNE T. (2018). Corpora with part-of-speech annotations for three regional languages of France : Alsatian, Occitan and Picard. In *International Conference on Language Resources and Evaluation*, Miyazaki, Japan. HAL : hal-02358018.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146. DOI : 10.1162/tacl_a_00051.

BRAS M. & VERGEZ-COURET M. (2016). BaTelÒc : a text base for the Occitan language. In V. FERREIRA & P. BOUDA, Éds., *Language Documentation and Conservation in Europe*, p. 133–149 : Honolulu : University of Hawaï Press. HAL : hal-00987241.

BRAS M., VERGEZ-COURET M., HATHOUT N., SIBILLE J., SÉGUIER A. & DAZÉAS B. (2020). Lofl̀oc : Lexic obèrt flechit occitan. In J.-F. COUROUAU, Éd., *Fidélités et dissidences (Actes du XIIe congrès de l'Association Internationale d'Études Occitanes)*, Albi : Centre d'Etude de la Littérature Occitane.

CORRAL A., LETURIA I., SÉGUIER A., BARRET M., DAZÉAS B., BOULA DE MAREÜIL P. & QUINT N. (2020). Neural text-to-speech synthesis for an under-resourced language in a diglossic environment : the case of Gascon Occitan. In *Proceedings of the 1st Joint SLTU (Spoken Language Technologies for Under-resourced languages) and CCURL (Collaboration and Computing for Under-Resourced Languages) Workshop «Language Resources and Evaluation Conference–Marseille–11–16 May 2020»*, p. 53–60 : European Language Resources Association (ELRA).

COSTA-JUSSÀ M. R., CROSS J., ÇELEBI O., ELBAYAD M., HEAFIELD K., HEFFERNAN K., KALBASSI E., LAM J., LICHT D., MAILLARD J. *et al.* (2022). No language left behind : Scaling human-centered machine translation. *arXiv preprint arXiv :2207.04672*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

GESMUNDO A. & SAMARDZIC T. (2012). Lemmatising serbian as category tagging with bidirectional sequence classification. In N. C. C. CHAIR), K. CHOUKRI, T. DECLERCK, M. U. DOĞAN, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey : European Language Resources Association (ELRA).

GOLDHAHN D., ECKART T. & QUASTHOFF U. (2012). Building large monolingual dictionaries at the Leipzig corpora collection : From 100 to 200 languages. In N. CALZOLARI, K. CHOUKRI, T. DECLERCK, M. U. DOĞAN, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey : European Language Resources Association (ELRA).

GRZEGORZ CHRUPALA G. D. & VAN GENABITH J. (2008). Learning morphology with morfette. In B. M. J. M. J. O. S. P. D. T. NICOLETTA CALZOLARI (CONFERENCE CHAIR), KHALID CHOUKRI, Éd., *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco : European Language Resources Association (ELRA). http ://www.lrec-conf.org/proceedings/lrec2008/.

MANJAVACAS E., KÁDÁR Á. & KESTEMONT M. (2019). Improving lemmatization of non-standard languages with joint learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 1493–1503, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1153.

MILETIC A., BERNHARD D., BRAS M., LIGOZAT A.-L. & VERGEZ-COURET M. (2019a). Transformation d'annotations en parties du discours et lemmes vers le format universal dependencies : étude de cas pour l'alsacien et l'occitan. Poster, HAL : hal-02123743.

MILETIC A., BRAS M., ESHER L., SIBILLE J. & VERGEZ-COURET M. (2019b). Building a treebank for Occitan : what use for Romance UD corpora ? In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, p. 2–11, Paris, France : Association for Computational Linguistics. DOI : 10.18653/v1/W19-8002.

MILETIC A., BRAS M., VERGEZ-COURET M., ESHER L., POUJADE C. & SIBILLE J. (2020a). Building a Universal Dependencies Treebank for Occitan. In *Proceedings of The 12th Language Resources and Evaluation Conference*, p. 2932–2939, Marseille, France : European Language Resources Association.

MILETIC A., BRAS M., VERGEZ-COURET M., ESHER L., POUJADE C. & SIBILLE J. (2020b). A four-dialect treebank for Occitan : Building process and parsing experiments. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, p. 140–149, Barcelona, Spain (Online) : International Committee on Computational Linguistics (ICCL).

MILETIĆ A. & SIEWERT J. (2023). Lemmatization experiments on two low-resourced languages : Low Saxon and Occitan. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, p. 163–173, Dubrovnik, Croatia : Association for Computational Linguistics.

MILETIĆ A. & SCHERRER Y. (2022a). OcWikiDisc : a Corpus of Wikipedia Talk Pages in Occitan. DOI : 10.5281/zenodo.7079580.

MILETIĆ A. & SCHERRER Y. (2022b). OcWikiDisc : a corpus of Wikipedia talk pages in Occitan. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, p. 70–79, Gyeongju, Republic of Korea : Association for Computational Linguistics.

MÜLLER T., COTTERELL R., FRASER A. & SCHÜTZE H. (2015). Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 2268–2274, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1272.

ORTIZ SUÁREZ P. J., SAGOT B. & ROMARY L. (2019). Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In P. BAŃSKI, A. BARBARESI, H. BIBER, E. BREITENEDER, S. CLEMATIDE, M. KUPIETZ, H. LÜNGEN & C. ILIADI, Éds., *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019*, p. 9 – 16, Mannheim : Leibniz-Institut für Deutsche Sprache. DOI : 10.14618/ids-pub-9021.

POUJADE C. (en préparation). *La linguistique outillée à l'épreuve de la variation : Ressources et outils pour les parlers occitans de l'Ariège*. Thèse de doctorat, Université de Toulouse.

QI P., ZHANG Y., ZHANG Y., BOLTON J. & MANNING C. D. (2020). Stanza : A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*.

RAJMAN M., LECOMTE J. & PAROUBEK P. (1997). *Format de description lexicale pour le français. Partie 2 : Description morpho-syntaxique*. Rapport interne, EPFL & INaLF. GRACE GTR-3-2.1.

SIBILLE J. (2002). Ecrire l'occitan : essai de présentation et de synthèse. In D. CAUBET, S. CHAKER & J. SIBILLE, Éds., *Les langues de France et leur codification. Ecrits divers – Ecrits ouverts*, Paris, France : Inalco / Association Universitaire des Langues de France L'Harmattan. HAL : hal-01296986.

VAN DER GOOT R., ÜSTÜN A., RAMPONI A., SHARAF I. & PLANK B. (2021). Massive choice, ample tasks (MaChAmp) : A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : System Demonstrations*, p. 176–197, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.eacl-demos.22.

VERGEZ-COURET M. (2016). *Description du lexique Lofl̀oc*. Research report, CLLE-ERSS. HAL : hal-01338774.

VERGEZ-COURET M. (2019). Tokenization for occitan (gascon and lengadocian). DOI : 10.5281/zenodo.2533873.

VERGEZ-COURET M. & URIELI A. (2015). Analyse morphosyntaxique de l'occitan languedocien : l'amitié entre un petit languedocien et un gros catalan. In *TALARE 2015*, Caen, France.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Stratégies d'apprentissage actif pour la reconnaissance d'entités nommées en français

Marco Naguib[1] Aurélie Névéol[1] Xavier Tannier[2]
(1) Université Paris-Saclay, CNRS, LISN, 91405 Orsay cedex, France
(2) Sorbonne Université, Inserm, Université Sorbonne Paris Nord, LIMICS, 75006 Paris, France
marco.naguib@lisn.upsaclay.fr, aurelie.neveol@lisn.upsaclay.fr,
xavier.tannier@sorbonne-universite.fr

RÉSUMÉ

L'annotation manuelle de corpus est un processus coûteux et lent, notamment pour la tâche de reconnaissance d'entités nommées. L'apprentissage actif vise à rendre ce processus plus efficace, en sélectionnant les portions les plus pertinentes à annoter. Certaines stratégies visent à sélectionner les portions les plus représentatives du corpus, d'autres, les plus informatives au modèle de langage. Malgré un intérêt grandissant pour l'apprentissage actif, rares sont les études qui comparent ces différentes stratégies dans un contexte de reconnaissance d'entités nommées médicales. Nous proposons une comparaison de ces stratégies en fonction des performances de chacune sur 3 corpus de documents cliniques en langue française : MERLOT, QuaeroFrenchMed et E3C. Nous comparons les stratégies de sélection mais aussi les différentes façons de les évaluer. Enfin, nous identifions les stratégies qui semblent les plus efficaces et mesurons l'amélioration qu'elles présentent, à différentes phases de l'apprentissage.

ABSTRACT

Sampling strategies in active learning for named entity recognition in French

Manual corpus annotation for NLP can be labor intensive and expensive. Active learning aims to achieve high accuracy with fewer training data by allowing a model to select the data to be annotated and used for learning. Some sampling strategies aim to select the most representative instances, while others aim to capture the instances that are most informative for the language model. Despite a growing interest in active learning in recent years, few studies provide a thorough comparison between those strategies in the context of medical named entity recognition. In this work, we apply these stategies on 3 French corpora in the clinical domain : MERLOT, QuaeroFrenchMed, and E3C. We provide an extensive comparison of those strategies and discuss various ways of evaluating them. Finally we determine those which seem most effective, and measure the estimated improvement they can provide at different stages of the training process.

MOTS-CLÉS : Reconnaissance d'entités nommées ; Documents cliniques ; Apprentissage actif.

KEYWORDS: Named entity recognition ; Clinical narratives ; Active learning.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 232–247.

1 Introduction

L'apprentissage supervisé repose sur l'hypothèse de la disponibilité de données annotées de haute qualité. Or, la collecte et l'annotation de telles données peuvent s'avérer coûteuses en ressources et en temps (Fort *et al.*, 2012; Grouin *et al.*, 2014). Ceci est particulièrement vrai dans le domaine des textes cliniques, où la tâche d'annotation doit être confiée à des personnes faisant preuve d'un haut niveau d'expertise (Campillos *et al.*, 2017). La question d'efficacité en données est donc primordiale. L'*active learning* (Lewis & Gale, 1994) ou apprentissage actif propose d'augmenter l'efficacité d'un algorithme d'apprentissage supervisé, en lui permettant d'interagir directement avec la source de données (souvent l'annotateur). On l'oppose à un algorithme d'apprentissage supervisé, dit « passif », où l'intégralité de la supervision, à savoir, des données annotées servant au processus d'apprentissage, est disponible avant l'exécution de l'algorithme. L'*active learning* propose, quant à lui, de faire intervenir l'algorithme d'apprentissage dans le processus de sélection des données à annoter, avant qu'elles ne le soient. L'objectif étant de parvenir à sélectionner les données les plus pertinentes pour l'apprentissage. Ici, nous nous intéressons au schéma d'*active learning* dit *pool-based* (Cheng *et al.*, 2013), dans lequel les données non annotées sont toutes disponibles avant l'apprentissage, et l'algorithme d'apprentissage cherche à estimer la pertinence de chaque portion de données. Ceci imite le contexte d'usage clinique, où disposer de bases de données non annotées de taille raisonnable peut se montrer bien plus facile que d'annoter ces données (Névéol *et al.*, 2014).

Si l'*active learning* connaît du succès dans le domaine de l'image ou la classification de texte, il n'y a pas de façon standard de l'employer, ni de l'évaluer, dans le contexte des tâches de prédiction structurée, comme la reconnaissance d'entités nommées. En particulier, de nombreuses stratégies de sélection d'exemples existent mais aucune d'entre elles ne s'impose, faute de succès. De plus, malgré l'intérêt grandissant pour l'*active learning*, les études l'appliquant à la reconnaissance d'entités nommées sont rares. Par exemple, la dernière en date s'intéressant au français est Claveau & Kijak (2015). Nous nous intéressons ici à appliquer, combiner et comparer quelques unes de ces stratégies, sur trois corpus de documents cliniques en langue française : MERLOT (Campillos *et al.*, 2017), QuaeroFrenchMed (Névéol *et al.*, 2014) et E3C (Magnini *et al.*, 2021). Les contributions de cet article sont les suivantes :

— Nous fournissons une comparaison approfondie des stratégies classiques de sélection d'exemples dans le domaine médical.
— Nous discutons de différentes façon d'évaluer l'*active learning* dans le cadre de la reconnaissance d'entités nommées.
— Nous montrons que deux stratégies simples basées sur la similarité de vocabulaire se révèlent les plus efficaces, et nous mesurons l'amélioration apportée par celles-ci.
— Nous mettons à disposition l'ensemble des scripts utilisés dans nos expériences [1].

2 État de l'art

L'*active learning* a été étudié depuis plus de vingt ans, (Cohn *et al.*, 1994a,b; Lewis & Catlett, 1994; Lewis & Gale, 1994). Zhang *et al.* (2022b) observent que si le nombre de publications étudiant l'*active learning* a connu un pic entre 2008 et 2010, cet intérêt semble avoir diminué entre 2011 et 2019, années de l'essor du *deep learning*. Plus récemment, on observe un regain d'intérêt

1. https://github.com/marconaguib/active-nlstruct

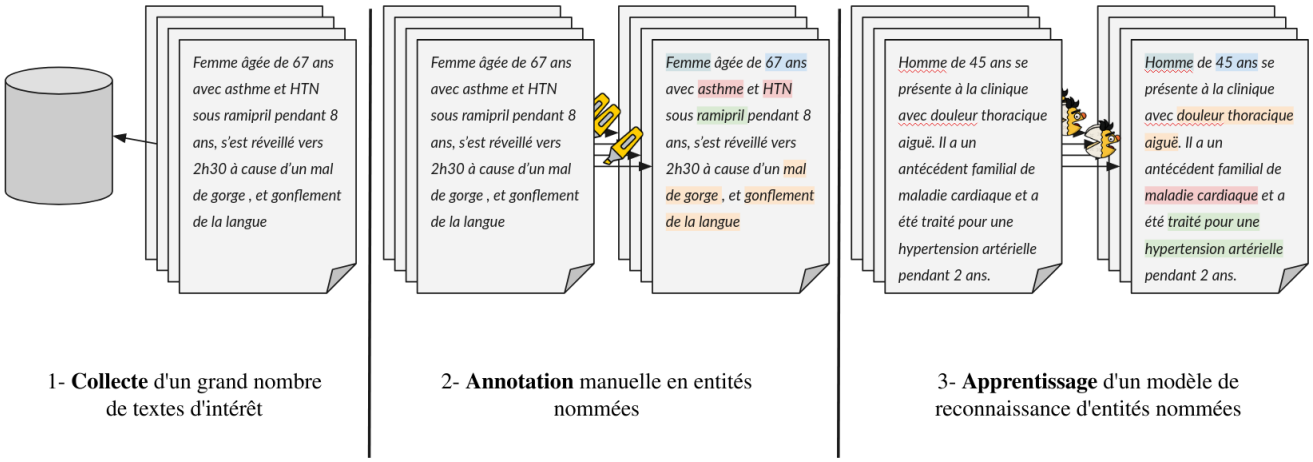


FIGURE 1 – Apprentissage supervisé (« passif »)

pour l'*active learning* qui se traduit dans l'*ACL anthology* par une montée du nombre d'articles l'étudiant en 2020. Ce regain d'intérêt pour l'*active learning* vise essentiellement à le combiner au *deep learning* (Ren *et al.*, 2021; Zhan *et al.*, 2022), et ainsi à rendre l'apprentissage des modèles plus efficace.

Une boucle de sélection, d'annotation et d'apprentissage Pour réaliser une tâche comme la reconnaissance d'entité nommées, la solution que propose généralement le *deep learning* est l'apprentissage supervisé. Elle consiste en 3 grandes étapes (cf. figure 1). A l'étape 1, on collecte un grand nombre de textes du domaine d'intérêt. L'étape 2 consiste à annoter manuellement ces textes en entité nommées. Cette étape peut selon les domaines nécessiter plus ou moins de compétence, et peut ainsi s'avérer plus ou moins coûteuse. Enfin, cette base de données ainsi annotée sert pour l'apprentissage d'un modèle de reconnaissance d'entités nommées à l'étape 3.

L'*active learning* part de l'intuition que tous les textes à annoter n'ont pas la même pertinence. Ainsi, il vise à sélectionner ceux qui sont les plus pertinents à annoter, pour réduire ce coût d'annotation et augmenter l'efficacité de l'entraînement. Il propose ainsi de remplacer les étapes 2 et 3 par une boucle (cf. figure 2) où l'on sélectionne d'abord un petit nombre d'exemples qu'on estime les plus pertinents à annoter (2a). Ceux-ci sont ensuite annotés (2b) et utilisés pour démarrer l'entraînement (2c), avant de sélectionner à nouveau un petit nombre d'exemples etc.

Stratégies de sélection Il existe de nombreuses façon d'estimer la pertinence des exemples (et ainsi sélectionner ceux qui sont les plus pertinents). On peut distinguer deux familles de stratégies de sélection.

1. Les stratégies d'**informativité** visent à repérer les exemples qui semblent les plus informatifs pour le modèle.

   Il est courant d'estimer l'informativité de chaque exemple par l'**incertitude** qu'exprime le modèle face à celui-ci. Ainsi, les exemples les plus pertinents à annoter seraient ceux pour lesquelles le modèle est le plus incertain (Lewis & Gale, 1994). Dans un modèle de classification probabiliste, appliquer cette méthode revient à calculer la distribution de probabilité pour chaque exemple sur les différentes classes possibles, puis de trier les exemples selon l'entropie de cette distribution (Tang *et al.*, 2002; Chen *et al.*, 2006; Zhu & Hovy, 2007),

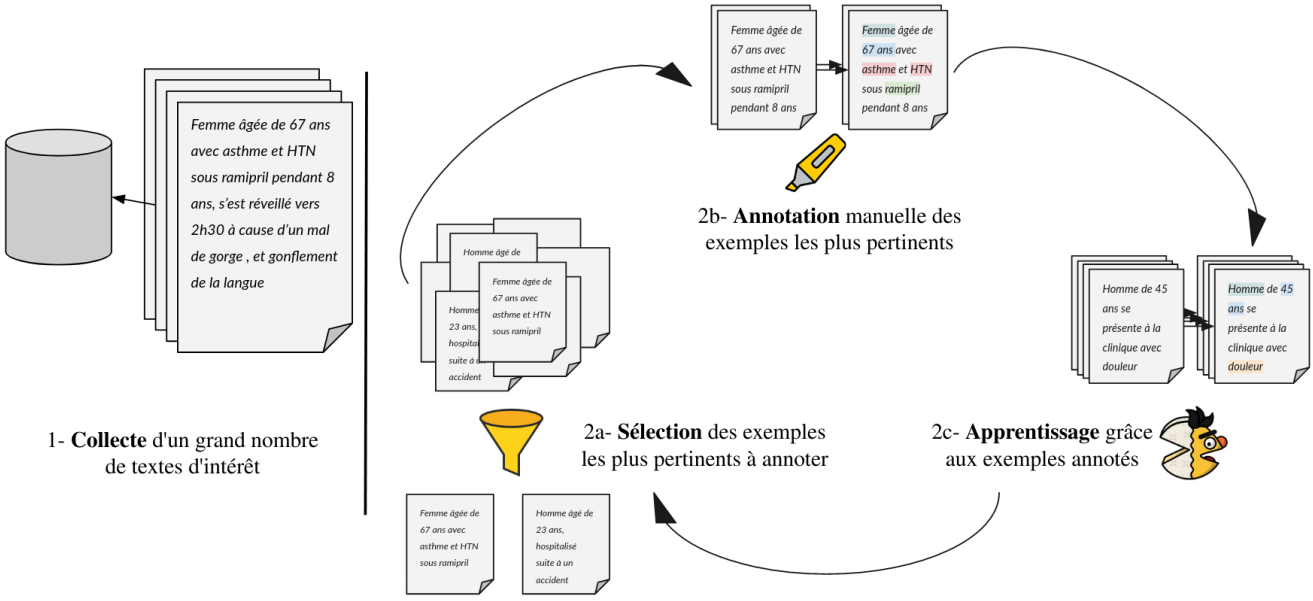


FIGURE 2 – Aprentissage actif

selon la probabilité attribuée à la classe la plus probable (Lewis & Gale, 1994; Culotta & McCallum, 2005) ou selon l'écart de probabilité entre les deux classes les plus probables (Scheffer *et al.*, 2001; Schein & Ungar, 2007).

Il n'existe pas de façon standard de **transposer** cette approche d'incertitude à la reconnaissance d'entités nommées. En effet, cette dernière associe à chaque séquence d'entrée (des tokens) une séquence de prédictions, ayant chacun une incertitude séparée. Ainsi, il faut agréger ces incertitudes en une incertitude globale à l'échelle de l'exemple (phrase ou document), pour pouvoir décider de l'informativité et ainsi la pertinence globale de celle-ci. Culotta & McCallum (2005) proposent de calculer d'abord la « confiance » du modèle en une prédiction $c_i = 1 - u_i$ où u_i est son incertitude pour le $i^{ème}$ élément de la séquence d'entrée, puis de calculer l'incertitude globale $\mathcal{I} = 1 - \Pi_{i=1}^{n} c_i$. Cette méthode a tendance à préférer les séquences longues. Tang *et al.* (2002) et Shen *et al.* (2018) proposent en revanche de moyenner les incertitudes pour les normaliser par la taille de la séquence.

Il existe par ailleurs d'autres approches pour estimer l'informativité. La **divergence locale** consiste à examiner les prédictions du modèle dans la région locale de chaque exemple grâce à la recherche des voisins les plus proches (Margatina *et al.*, 2021), à des perturbations locales (Zhang *et al.*, 2022a) ou à l'augmentation de données (Jiang *et al.*, 2020). Le **désaccord multi-modal** (Shen *et al.*, 2018; McCallum & Nigam, 1998; Houlsby *et al.*, 2011) vise, lui, à entraîner un « comité » de modèles et se servir du désaccord entre eux.

2. Les stratégies de **représentativité** visent à prendre en compte la similarité des exemples entre eux, pour éviter la redondance et la sélection d'intrus, problèmes auxquels les approches d'informativité peuvent être vulnérables (Roy & McCallum, 2001; Karamcheti *et al.*, 2021).

   Par exemple, l'approche de **densité** vise à sélectionner les exemples qui présentent un vocabulaire le plus similaire en moyenne à celui de tous les autres exemples. Cette similarité peut être mesurée par la fréquence de mots ou n-grams (McCallum & Nigam, 1998; Settles & Craven, 2008; Zhao *et al.*, 2020)

   L'approche de **diversité**, elle, vise à éviter la redondance, en sélectionnant des exemples variés qui représentent la variété de l'ensemble des entrées. Ainsi, l'on peut présenter des

exemples qui présentent le moins de similarité entre eux (Brinker, 2003), et/ou le moins de similarité avec les données déjà annotées (Eck *et al.*, 2005; Bloodgood & Callison-Burch, 2010; Erdmann *et al.*, 2019). Cette sélection peut se faire de façon itérative (Shen *et al.*, 2004; Geifman & El-Yaniv, 2017; Sener & Savarese, 2018) ou on employant la classification non supervisée (Zhdanov, 2019; Yu *et al.*, 2022).

3. Ces deux familles d'approches ne sont pas incompatibles et l'on peut les **combiner** de plusieurs façons. Par exemple, Chen *et al.* (2011) proposent de calculer les « scores de pertinence » de chaque exemple, puis de calculer simplement une somme pondérée de ces scores. Mirroshandel & Nasr (2011) et Tang *et al.* (2002) appliquent d'abord une première stratégie pour sélectionner un sous ensemble des données, puis en utilisent un autre pour sélectionner les exemples. On peut également procéder à des combinaisons dynamiques. Ambati *et al.* (2011) et Wu *et al.* (2017) proposent de reposer d'abord sur une approche de représentativité, puis, au fur et à mesure de l'apprentissage, passer à une approche d'informativité.

Dans ce travail, nous examinons des stratégies d'incertitude, de densité et de diversité, ainsi qu'une concaténation simple de ces deux dernières, dans le contexte de la reconnaissance d'entités nommées médicales.

3 Expérimentation

Corpus utilisés Dans ce travail, nous étudions le comportement de l'*active learning* sur trois corpus cliniques annotés en entités nommées. Le tableau 1 indique les différent types d'entités dans chaque corpus, ainsi que leurs nombres.

1. **MERLOT** (Campillos *et al.*, 2017) est un corpus privé de documents cliniques pseudonymisés collectés d'hôpitaux français. Ils couvrent différents types de documents : compte-rendus de prise en charge, compte-rendus post-opératoires, et courriers dans le domaine de la gastro-entérologie. Le corpus est annoté en 21 types d'entités nommées. Cependant, nous choisissons de suivre les remplacements indiqués par Bannour *et al.* (2022), pour n'obtenir ainsi que 15 types d'entités.
2. **QuaeroFrenchMed** (Névéol *et al.*, 2014) est un corpus composé de deux parties que nous traitons séparément. La première partie, **EMEA** est une collection de 13 notices patient concernant des médicaments commercialisés en Europe, fournis par l'Agence Européenne des Médicaments. La seconde partie **MEDLINE**, consiste en 2500 titres d'articles scientifiques indexés dans la base de données MEDLINE [2]. Ces deux parties sont annotées en 10 types d'entités nommées.
3. **E3C** (Magnini *et al.*, 2021) est un corpus européen de cas cliniques. Nous utilisons la partie française, composée de 1615 cas cliniques collectionnés du domaine public : articles scientifiques indexés sur PubMed [2] et articles scientifiques sous license CC-by. Il est annoté en 6 types d'entités nommées cliniques, dont un type : entité clinique (`CLINENTITY`) que nous faisons le choix de désagréger en sous-types pour avoir un schéma d'annotation avec une diversité se rapprochant de celui des autres corpus. Chaque entité de ce type est annotée par un attribut `CUI` qui désigne l'identifiant unique du concept associé dans le métathésaurus UMLS (Unified Medical Language System). Grâce à cet identifiant, nous pouvons ainsi récupérer le type sémantique (McCray *et al.*, 2001) correspondant (pathologie, symptôme...).

2. http://pubmed.ncbi.nlm.nih.gov/

MERLOT	EMEA	MEDLINE	E3C
ANAT (4449)	ANAT (583)	ANAT (1499)	ACTOR (427)
CHEM (1374)	CHEM (2482)	CHEM (1055)	Acquired_Abnormality (15)
Concept_Idea (2964)	DEVI (170)	DEVI (128)	Anatomical_Abnormality (25)
DEVI (1068)	DISO (1510)	DISO (2843)	Bacterium (1)
DISO (4593)	GEOG (65)	GEOG (131)	BODYPART (659)
DOSE (928)	LIVB (817)	LIVB (941)	Cell_or_Molecular_Dysfunction (7)
Genes_Proteins (5)	OBJC (174)	OBJC (100)	Congenital_Abnormality (11)
Hospital (806)	PHEN (62)	PHEN (158)	Disease_or_Syndrome (374)
LIVB (3921)	PHYS (344)	PHYS (469)	EVENT (1100)
Localization (840)	PROC (952)	PROC (1750)	Finding (178)
MEAS (5322)			Injury_or_Poisoning (24)
MODE (252)			Mental_or_Behavioral_Dysfunction (22)
PHEN (905)			Neoplastic_Process (99)
PROC (8291)			Pathologic_Function (149)
TEMP (3940)			RML (508)
			Sign_or_Symptom (179)
			TIMEX3 (333)
			Virus (1)

TABLE 1 – Types d'entités présents dans les différents corpus

En appliquant l'*active learning* au traitement automatique des langues, se pose la question de l'unité documentaire considérée, et donc du découpage du texte. Même si les corpus sont naturellement découpés par document (sauf MEDLINE où chaque titre est un document séparé), il est plus courant dans l'*active learning* de considérer un découpage par phrase (Chen *et al.*, 2015). En effet, toutes les phrases d'un document n'ont pas la même pertinence, et on peut préférer en sélectionner certaines et pas d'autres. De plus, les approches d'informativité se basent souvent sur les représentations et les prédictions faites par le modèle pour chaque mot. Considérer les documents dans leur globalité pourrait donc noyer l'information que le modèle peut émettre à l'échelle des mots. C'est d'ailleurs ce qui encourage Radmard *et al.* (2021) à découper les corpus en n-grammes et évaluer l'informativité de chacun. Nous en restons à un découpage par phrases pour des raisons d'efficacité. Nous procédons donc à ce découpage pour MERLOT, EMEA et QuaeroFrenchMed, grâce à une expression régulière permettant une séparation aux ponctuations fortes et aux multiples sauts à la lignes.

Stratégies de sélection Dans ce travail, nous examinons et comparons plusieurs stratégies de sélection comme suit.

— `random` sélectionne aléatoirement des exemples.

— `common_vocab` identifie les 500 n-grammes les plus fréquents dans le corpus, puis trie les exemples selon le nombre de n-grammes fréquents qu'ils contiennent. Elle est inspirée de Zhao *et al.* (2020), qui l'appliquent à la tâche de traduction.

— `diverse_vocab` sélectionne un exemple de façon aléatoire, puis, itérativement, sélectionne l'exemple qui présente le plus de *n-grams* non déjà vus dans les exemples déjà sélectionnés. Elle est inspirée de Kirsch *et al.* (2019).

— `diverse_pred` sélectionne un exemple de façon aléatoire, puis, itérativement, sélectionne l'exemple qui présente le plus de types d'entités prédites, non déjà prédites dans les exemples précédemment sélectionnés.

— `uncertainty_mean_min3` calcule les confiances du modèle pour chaque entité prédite [3], puis trie les exemples selon la confiance moyenne croissante. Pour éviter les phrases « trop courtes », on restreint le choix aux exemples présentant plus de 3 entités prédites. Elle

3. Pour mieux tenir compte des entités imbriquées, notre modèle classifie chaque partie (chaque *span*) de l'exemple. Ainsi, ce sont les confiances en ces prédictions qu'on moyenne sur l'exemple et qu'on trie.

est inspirée de Shen *et al.* (2018).

Il est important de noter que `common_vocab` et `diverse_vocab` ne dépendent pas des prédictions du modèle, et peuvent donc être appliquées avant l'entraînement. `diverse_pred` et `uncertainty_mean_min3`, en revanche, nécessitent une inférence sur toutes les données non-annotées du corpus, l'annotateur doit donc attendre l'entraînement du modèle. En pratique cependant, on peut imaginer un scénario où l'annotateur a une longueur d'avance par rapport à l'entraînement du modèle : à la première étape, le modèle sélectionne 2 *batchs* d'exemples à annoter, l'annotateur annote d'abord le premier *batch* et le soumet, puis, pendant qu'il annote le deuxième *batch*, le modèle commence l'entraînement sur le premier, etc.

Déroulement de la simulation Pour examiner ces stratégies, nous simulons une boucle d'*active learning*. Pour ce faire, le jeu d'entraînement est initialisé à l'ensemble vide, puis, à chaque itération, on y ajoute seulement les documents sélectionnés par la stratégie en question.

Afin de ne pas multiplier les expériences, nous utilisons un unique modèle pour évaluer toutes ces stratégies, NLStruct [4]. Il s'agit d'un Bi-LSTM-CRF, combiné à un modèle de langue CamemBERT-base (Martin *et al.*, 2020), il est décrit en détail par Wajsbürt (2021). Nous utilisons tous les hyperparamètres proposés par défaut.

Nous fixons le nombre de phrases annotées à chaque itération à 10. Ce qui correspond par exemple à 10-20 minutes en moyenne dans le corpus MEDLINE. À chaque itération, la stratégie en question sélectionne donc 10 phrases du corpus, et l'on simule l'annotation manuelle en dévoilant les annotations *gold standard* concernant ces phrases. On applique d'abord cette stratégie $k = 2$ fois pour choisir 20 phrases comme jeu de validation. Puis pendant 10 itérations, on intègre les phrases annotées dans le jeu d'entraînement et entraîne le modèle sur l'ensemble des données annotées. Nous discutons de ce choix de 10 itérations dans la partie 5. Chaque entraînement consiste en 1000 étapes d'optimisation avec arrêt prématuré si aucune amélioration sur le jeu de validation n'est observée pendant 300 étapes.

Mesures Nous mesurons les performances du modèle en fonction de l'effort à fournir par l'annotateur au fil des itérations. Afin de mesurer précisément l'effort annotateur, nous choisissons de rapporter **le nombre de mots annotés** (Chen *et al.*, 2015). Quant à la mesure de performance, nous rapportons la mesure classique du score f_1 **toutes classes confondues**, que nous appelons f_1^{micro}. De plus, pour tenir compte de la performance sur les classes rares, nous rapportons également la **moyenne simple des scores** f_1 obtenus sur chaque classe séparément, sans pondération. Nous l'appelons f_1^{macro}. La figure 3 rapporte l'évolution de ces 2 scores au fil de l'apprentissage. Chaque courbe représente la moyenne du score sur 3 graines aléatoires utilisées, dans un intervalle de confiance à 95%.

Nous rapportons également dans le tableau 2 pour chaque corpus c et stratégie s une quantité que nous appelons **Performance relative à** 1000 **mots** ou $\mathcal{P}_{1000}(c, s)$. Cette quantité mesure le ratio entre un score de performance obtenu en entraînant un modèle sur au moins 1000 mots du corpus c choisis selon la stratégie s, et celui obtenu en entraînant le modèle sur l'ensemble de c.

4. https://github.com/percevalw/nlstruct

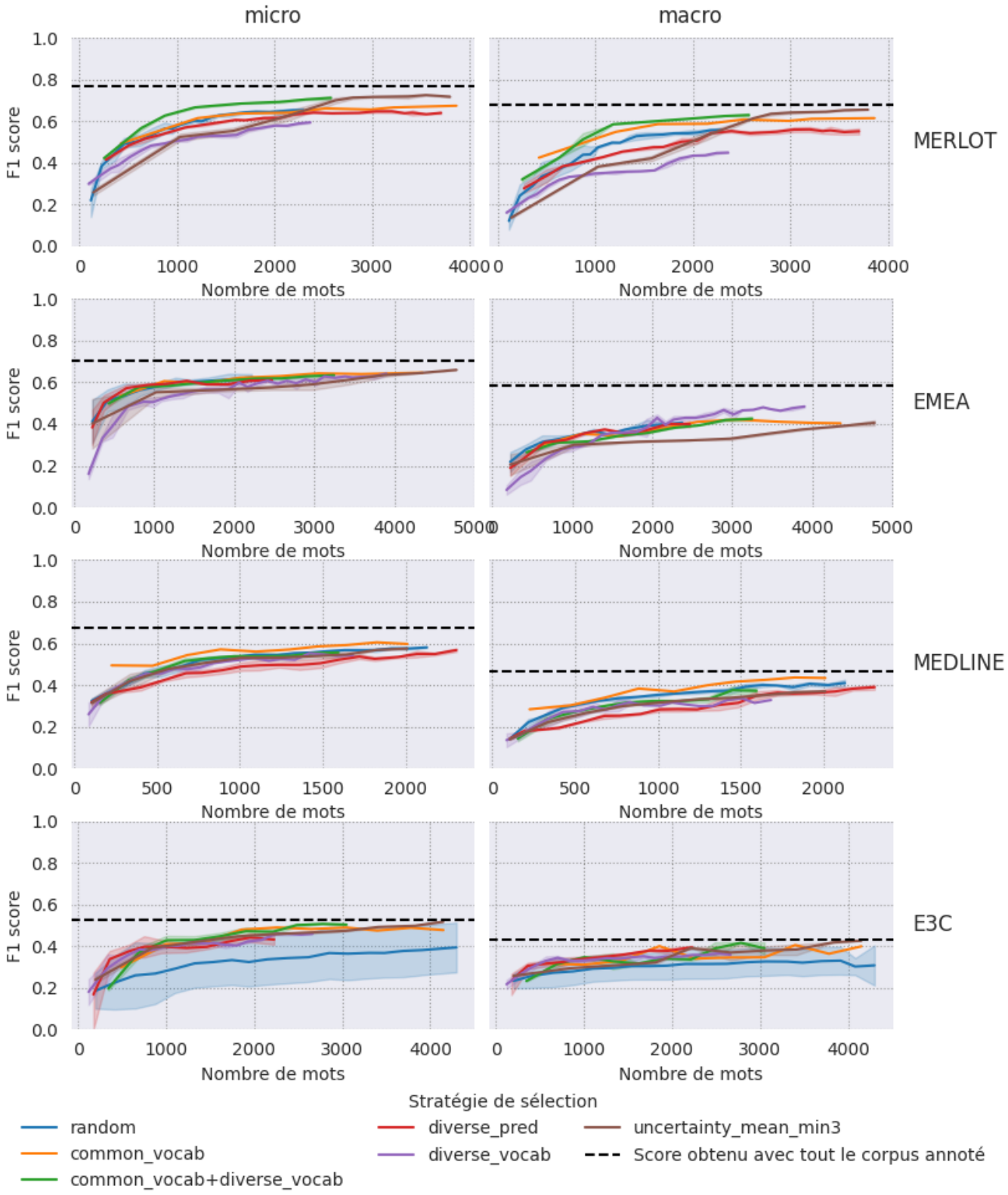


FIGURE 3 – Courbes d'évolution de f_1^{micro} et f_1^{macro} en fonction du nombre de mots annotés. Noter que les courbes ne s'arrêtent pas toutes à la même abscisse. En effet, les stratégies sélectionnent des séquences de tailles différentes. `common_vocab+diverse_pred` désigne la concaténation des deux stratégies : chacune d'entre elles sélectionne 5 exemples.

Stratégie	Corpus							
	MERLOT		EMEA		MEDLINE		E3C	
`random`	0.76	(0.70)	0.83	(0.60)	0.81	(0.77)	0.56	(0.67)
`common_vocab`	0.80	(0.81)	**0.86**	(0.60)	**0.83**	(**0.79**)	0.79	(0.76)
`diverse_vocab`	0.69	(0.61)	0.74	(0.50)	0.78	(0.64)	0.76	(0.76)
`diverse_pred`	0.71	(0.58)	0.85	(**0.61**)	0.73	(0.60)	0.74	(**0.81**)
`common_vocab+diverse_vocab`	**0.87**	(**0.86**)	0.83	(0.54)	0.80	(0.69)	**0.82**	(0.80)
`uncertainty_mean_min3`	0.68	(0.56)	0.79	(0.51)	0.79	(0.68)	0.80	(0.71)

TABLE 2 – Performances relatives à 1000 mots selon f_1^{micro} (f_1^{macro}).

4 Résultats

De prime abord, nous remarquons que des performances raisonnables peuvent être atteintes avec peu de phrases annotées, même quand celles-ci sont tirées aléatoirement. Par exemple, 1000 mots tirés aléatoirement du corpus MERLOT (<1 % du corpus) et annotés sont suffisants pour atteindre 0,61 de score f_1^{micro}, soit 70 % du score atteint en entraînant le même modèle sur l'intégralité du corpus annoté. On observe ce phénomène dans toutes les applications similaires, mais il semble être accentué par la redondance particulière aux documents cliniques (Cohen *et al.*, 2013; Searle *et al.*, 2021). Effectivement, le corpus MEDLINE, composé de titres d'articles scientifiques (on peut donc penser qu'il est particulèrement peu redondant) présente une performance relative à 1000 mots de 77 %. De manière générale, nous observons que dans chaque graphique, une ou plusieurs stratégies ont une meilleure évolution que `random`. Cependant, il n'y a pas une stratégie qui semble la meilleure pour tous les corpus. La stratégie `common_vocab` semble tout de même souvent efficace. Il est intéressant de voir qu'une simple concaténation des méthodes `common_vocab` et `diverse_vocab` peut dans certains cas avoir des meilleurs résultats que chacune d'elles séparément.

Nous observons par ailleurs que la stratégie `uncertainty_mean_min3` qui est la plus adoptée dans la littérature (Chen *et al.*, 2015; Shen *et al.*, 2018) ne semble pas adaptée au contexte de peu de phrases annotées dans lequel nous nous plaçons (cf. partie 5).

Une grande majorité des travaux sur l'*active learning* (Chen *et al.*, 2015; Shen *et al.*, 2018; Liu *et al.*, 2022; Radmard *et al.*, 2021) font le choix de cadrer les graphiques d'évolution de performance sur la partie supérieure, afin de mieux visualiser l'écart entre les performances des différentes méthodes. Ici, nous faisons le choix de les cadrer entre 0 et 1, ce qui permet de voir le caractère marginal des améliorations obtenues grâce aux meilleures stratégies.

5 Discussion et perspectives

Mesure de l'effort Il est courant d'estimer l'effort de l'annotateur par le nombre de phrases ou de mots annotés. Mais certains types d'entités peuvent être plus difficiles à annoter que d'autres. Notamment, les stratégies d'informativité visent précisément à sélectionner les exemples ambigus et difficile à annoter. Aussi, une comparaison juste des stratégies de sélection prendrait-elle cet effort en compte. Fort *et al.* (2012) fournissent une modélisation de cet effort en fonction du schéma d'annotation. C'est donc une perspective d'amélioration que nous considérons.

Une meilleure estimation de l'effort d'annotation peut même guider les stratégies de sélection. En

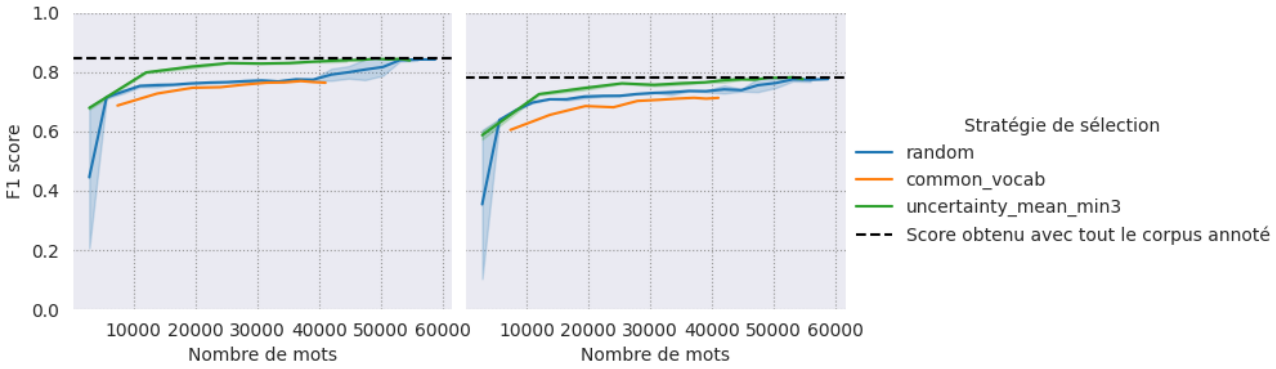


FIGURE 4 – Comparaison à grande échelle des différentes stratégies sur MERLOT

	Corpus			
Stratégie	MERLOT	EMEA	MEDLINE	E3C
`common_vocab`	<1	<1	<1	<1
`diverse_vocab`	3	2	2	<1
`diverse_pred`	105	45	70	55
`uncertainty_mean_min3`	103	42	68	55

TABLE 3 – Comparaison du temps d'exécution moyen en secondes des différentes stratégies

effet, certains travaux (Tomanek & Hahn, 2010; Wei *et al.*, 2019) développent des stratégies qui visent à estimer non seulement la pertinence mais aussi le coût d'annotation de chaque exemple, pour ensuite les trier selon ce que Haertel *et al.* (2008) appellent « retour sur investissement ».

Intérêt et budget d'annotation La plupart des travaux sur l'*active learning* l'étudient dans toutes ses phases (Shen *et al.*, 2018; Radmard *et al.*, 2021). Par exemple, ces derniers l'évaluent pour un nombre de mots annotés allant jusqu'à 1 million, pour l'anglais et le chinois. En revanche, nous nous sommes intéressés au contexte plus naturel d'une centaine de phrases annotées, ce qui vaut entre une et deux heures de travail pour l'annotateur. Pour pouvoir comparer notre implémentation à celles de l'état de l'art, nous procédons à une simulation d'*active learning* sur MERLOT (le plus grand de nos corpus) en fixant le nombre de phrases à annoter à chaque itération à 250 ($\approx$ 5 % du corpus). La figure 4 montre ainsi les courbes d'évolution, et l'on trouve en effet que `common_vocab` n'est plus très intéressante à grande échelle et `uncertainty_mean_min3` devient plus intéressante dans ce cadre. Elle apporte, en effet, des amélioration similaires à celles obtenues par Liu *et al.* (2022) et Zhou *et al.* (2021) Ainsi, nous pouvons émettre l'hypothèse qu'une stratégie de combinaison dynamique qui passe progressivement d'une stratégie de représentativité à une stratégie d'informativité mériterait une attention particulière, dans un prochain travail.

Temps d'attente Le tableau 3 montre la moyenne de temps d'exécution d'une requête de sélection pour chaque stratégie, mesurées sur une carte GeForce GTX 1080 Ti (11 Go). On peut observer que les stratégies qui requièrent une inférence sur l'ensemble des données annotées (à savoir `diverse_pred` et `uncertainty_mean_min3`) sont les plus longues. Ce n'est pas une surprise, mais ce paramètre est très rarement mentionné dans les travaux sur l'*active learning*. Que ce soit en termes de temps d'attente ou d'impact carbone, l'amélioration des performances étant relativement limitée, il est pourtant permis de remettre en question leur utilisation, au profit de stratégies basées sur le vocabulaire, qui ne nécessitent pas d'être relancées à chaque itération.

Classes rares L'*active learning* a fait l'objet d'études pour améliorer la performance sur les classes

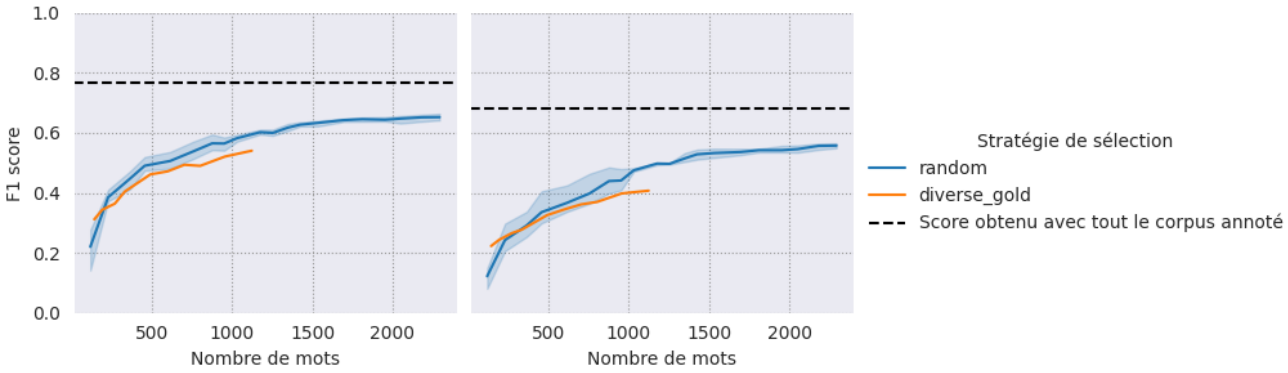


FIGURE 5 – Comparaison entre `random` et `diverse_gold` sur MERLOT

rares (Zhu & Hovy, 2007). Nous avons examiné une stratégie supplémentaire, `diverse_gold` qui imite ce contexte. Elle sélectionne un exemple de façon aléatoire, puis, itérativement, cherche les phrases qui maximisent le nombre d'apparition de l'entité la plus rarement sélectionnée. Cette méthode cherche donc à représenter toutes les entités de manière équitable et donc à sur-représenter les types rares. Les résultats obtenus n'ont pas été encourageants. À titre d'exemple, la figure 5 montre les performances de cette stratégie, comparée à `random`, sur MERLOT (cf. tableau 1 pour la distribution des types d'entités).

Stabilité Enfin, remarquons que les variations du modèle `random` sont parfois plus fortes que celles des autres stratégies (corpus E3C, et dans une moindre mesure EMEA et MERLOT). Certaines sélections aléatoires peuvent conduire à de très mauvais résultats, ce qui n'est pas le cas des stratégies d'*active learning*, beaucoup plus stables. Ce point peut prendre de l'importance lorsque la campagne d'annotation utilise une technique de pré-annotation des textes à partir d'un modèle entraîné avec peu de données, pour faciliter le travail humain.

Conclusion Nous avons étudié cinq stratégies d'*active learning* pour la reconnaissance d'entités nommées dans quatre corpus médicaux en français. Nos résultats suggèrent que les stratégies de représentativité sont particulièrement intéressantes sur des petits corpus en terme de temps de calcul et de stabilité des performances.

6 Remerciements

Nous remercions le Service d'Informatique Biomédicale (SIBM) ainsi que l'équipe CISMeF du CHU de Rouen qui nous ont permis d'utiliser le corpus LERUDI pour cette étude.

Références

AMBATI V., VOGEL S. & CARBONELL J. (2011). Multi-strategy approaches to active learning for statistical machine translation. In *Proceedings of Machine Translation Summit XIII : Papers*, Xiamen, China.

BANNOUR N., WAJSBÜRT P., RANCE B., TANNIER X. & NÉVÉOL A. (2022). Privacy-preserving mimic models for clinical named entity recognition in French. *Journal of Biomedical Informatics*, **130**, 104073. DOI : 10.1016/j.jbi.2022.104073, HAL : hal-03655039.

BLOODGOOD M. & CALLISON-BURCH C. (2010). Bucking the trend : Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 854–864, Uppsala, Sweden : Association for Computational Linguistics.

BRINKER K. (2003). Incorporating diversity in active learning with support vector machines. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, p. 59–66 : AAAI Press.

CAMPILLOS L., DELÉGER L., GROUIN C., HAMON T., LIGOZAT A.-L. & NÉVÉOL A. (2017). A French clinical corpus with comprehensive semantic annotations : development of the Medical Entity and Relation LIMSI annOtated Text corpus (MERLOT). *Language Resources and Evaluation*, **52**(2), 571–601. DOI : 10.1007/s10579-017-9382-y, HAL : hal-01631743.

CHEN C., PALMER A. & SPORLEDER C. (2011). Enhancing active learning for semantic role labeling via compressed dependency trees. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, p. 183–191, Chiang Mai, Thailand : Asian Federation of Natural Language Processing.

CHEN J., SCHEIN A., UNGAR L. & PALMER M. (2006). An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, p. 120–127, New York City, USA : Association for Computational Linguistics.

CHEN Y., LASKO T. A., MEI Q., DENNY J. C. & XU H. (2015). A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics*, **58**, 11–18. DOI : https ://doi.org/10.1016/j.jbi.2015.09.010.

CHENG Y., CHEN Z., LIU L., WANG J., AGRAWAL A. & CHOUDHARY A. (2013). Feedback-driven multiclass active learning for data streams. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, p. 1311–1320, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/2505515.2505528.

CLAVEAU V. & KIJAK E. (2015). Stratégies de sélection des exemples pour l'apprentissage actif avec des champs aléatoires conditionnels. In *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, p. 13–24, Caen, France : ATALA.

COHEN R., ELHADAD M. & ELHADAD N. (2013). Redundancy in electronic health record corpora : Analysis, impact on text mining performance and mitigation strategies. *BMC bioinformatics*, **14**, 10. DOI : 10.1186/1471-2105-14-10.

COHN D., GHAHRAMANI Z. & JORDAN M. (1994a). Active learning with statistical models. In G. TESAURO, D. TOURETZKY & T. LEEN, Éds., *Advances in Neural Information Processing Systems*, volume 7 : MIT Press.

COHN D. A., ATLAS L. E. & LADNER R. E. (1994b). Improving generalization with active learning. *Machine Learning*, **15**, 201–221.

CULOTTA A. & MCCALLUM A. (2005). Reducing labeling effort for structured prediction tasks. In *AAAI Conference on Artificial Intelligence*.

ECK M., VOGEL S. & WAIBEL A. (2005). Low cost portability for statistical machine translation based on n-gram frequency and TF-IDF. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

ERDMANN A., WRISLEY D. J., ALLEN B., BROWN C., COHEN-BODÉNÈS S., ELSNER M., FENG Y., JOSEPH B., JOYEUX-PRUNEL B. & DE MARNEFFE M.-C. (2019). Practical, efficient, and customizable active learning for named entity recognition in the digital humanities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 2223–2234, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1231.

FORT K., NAZARENKO A. & ROSSET S. (2012). Modeling the complexity of manual annotation tasks : a grid of analysis. In *Proceedings of COLING 2012*, p. 895–910, Mumbai, India : The COLING 2012 Organizing Committee.

GEIFMAN Y. & EL-YANIV R. (2017). Deep active learning over the long tail. *CoRR*, **abs/1711.00941**.

GROUIN C., LAVERGNE T. & NÉVÉOL A. (2014). Optimizing annotation efforts to build reliable annotated corpora for training statistical models. In *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop*, p. 54–58, Dublin, Ireland : Association for Computational Linguistics and Dublin City University. DOI : 10.3115/v1/W14-4907.

HAERTEL R. A., SEPPI K. D., RINGGER E. K. & CARROLL J. L. (2008). Return on investment for active learning. In *Proceedings of the NIPS workshop on cost-sensitive learning*, volume 72 : Citeseer.

HOULSBY N., HUSZAR F., GHAHRAMANI Z. & LENGYEL M. (2011). Bayesian active learning for classification and preference learning. *CoRR*, **abs/1112.5745**.

JIANG Z., GAO Z., DUAN Y., KANG Y., SUN C., ZHANG Q. & LIU X. (2020). Camouflaged Chinese spam content detection with semi-supervised generative active learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 3080–3085, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.279.

KARAMCHETI S., KRISHNA R., FEI-FEI L. & MANNING C. (2021). Mind your outliers ! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 7265–7281, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.564.

KIRSCH A., VAN AMERSFOORT J. & GAL Y. (2019). Batchbald : Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, **32**.

LEWIS D. D. & CATLETT J. (1994). Heterogeneous uncertainty sampling for supervised learning. In W. W. COHEN & H. HIRSH, Éds., *Machine Learning Proceedings 1994*, p. 148–156. San Francisco (CA) : Morgan Kaufmann. DOI : https ://doi.org/10.1016/B978-1-55860-335-6.50026-X.

LEWIS D. D. & GALE W. A. (1994). A sequential algorithm for training text classifiers. In B. W. CROFT & C. J. VAN RIJSBERGEN, Éds., *SIGIR '94*, p. 3–12, London : Springer London.

LIU M., TU Z., ZHANG T., SU T., XU X. & WANG Z. (2022). Ltp : A new active learning strategy for crf-based named entity recognition. *Neural Process. Lett.*, **54**(3), 2433–2454. DOI : 10.1007/s11063-021-10737-x.

MAGNINI B., ALTUNA B., LAVELLI A., SPERANZA M. & ZANOLI R. (2021). The e3c project : European clinical case corpus. *Language*, **1**(L2), L3.

MARGATINA K., VERNIKOS G., BARRAULT L. & ALETRAS N. (2021). Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 650–663, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.51.

MARTIN L., MULLER B., SUÁ REZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

MCCALLUM A. & NIGAM K. (1998). Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, p. 350–358, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

MCCRAY A. T., BURGUN A. & BODENREIDER O. (2001). Aggregating umls semantic types for reducing conceptual complexity. *Studies in health technology and informatics*, **84**(0 1), 216.

MIRROSHANDEL S. A. & NASR A. (2011). Active learning for dependency parsing using partially annotated sentences. In *Proceedings of the 12th International Conference on Parsing Technologies*, p. 140–149, Dublin, Ireland : Association for Computational Linguistics.

NÉVÉOL A., GROUIN C., LEIXA J., ROSSET S. & ZWEIGENBAUM P. (2014). The quaero french medical corpus : A ressource for medical entity recognition and normalization. *Proc of BioTextMining Work*, p. 24–30.

RADMARD P., FATHULLAH Y. & LIPANI A. (2021). Subsequence based deep active learning for named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 4310–4321, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.332.

REN P., XIAO Y., CHANG X., HUANG P.-Y., LI Z., GUPTA B. B., CHEN X. & WANG X. (2021). A survey of deep active learning. *ACM Comput. Surv.*, **54**(9). DOI : 10.1145/3472291.

ROY N. & MCCALLUM A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, p. 441–448, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

SCHEFFER T., DECOMAIN C. & WROBEL S. (2001). Active hidden markov models for information extraction. In F. HOFFMANN, D. J. HAND, N. ADAMS, D. FISHER & G. GUIMARAES, Éds., *Advances in Intelligent Data Analysis*, p. 309–318, Berlin, Heidelberg : Springer Berlin Heidelberg.

SCHEIN A. & UNGAR L. (2007). Active learning for logistic regression : An evaluation. *Machine Learning*, **68**, 235–265. DOI : 10.1007/s10994-007-5019-5.

SEARLE T., IBRAHIM Z., TEO J. & DOBSON R. (2021). Estimating redundancy in clinical text. *Journal of Biomedical Informatics*, **124**, 103938. DOI : https ://doi.org/10.1016/j.jbi.2021.103938.

SENER O. & SAVARESE S. (2018). Active learning for convolutional neural networks : A core-set approach. In *International Conference on Learning Representations*.

SETTLES B. & CRAVEN M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, p. 1070–1079, Honolulu, Hawaii : Association for Computational Linguistics.

SHEN D., ZHANG J., SU J., ZHOU G. & TAN C.-L. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, p. 589–es, USA : Association for Computational Linguistics. DOI : 10.3115/1218955.1219030.

SHEN Y., YUN H., LIPTON Z. C., KRONROD Y. & ANANDKUMAR A. (2018). Deep active learning for named entity recognition. In *International Conference on Learning Representations*.

TANG M., LUO X. & ROUKOS S. (2002). Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 120–127, Philadelphia, Pennsylvania, USA : Association for Computational Linguistics. DOI : 10.3115/1073083.1073105.

TOMANEK K. & HAHN U. (2010). A comparison of models for cost-sensitive active learning. In *Coling 2010 : Posters*, p. 1247–1255.

WAJSBÜRT P. (2021). *Extraction and normalization of simple and structured entities in medical documents*. Theses, Sorbonne Université. HAL : tel-03624928.

WEI Q., CHEN Y., SALIMI M., DENNY J. C., MEI Q., LASKO T. A., CHEN Q., WU S., FRANKLIN A., COHEN T. *et al.* (2019). Cost-aware active learning for named entity recognition in clinical text. *Journal of the American Medical Informatics Association*, **26**(11), 1314–1322.

WU F., HUANG Y. & YAN J. (2017). Active sentiment domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1701–1711, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1156.

YU Y., KONG L., ZHANG J., ZHANG R. & ZHANG C. (2022). AcTune : Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1422–1436, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.102.

ZHAN X., WANG Q., HUANG K.-H., XIONG H., DOU D. & CHAN A. B. (2022). A comparative survey of deep active learning. *arXiv preprint arXiv :2203.13450*.

ZHANG S., GONG C., LIU X., HE P., CHEN W. & ZHOU M. (2022a). ALLSH : Active learning guided by local sensitivity and hardness. In *Findings of the Association for Computational Linguistics : NAACL 2022*, p. 1328–1342, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.findings-naacl.99.

ZHANG Z., STRUBELL E. & HOVY E. (2022b). A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 6166–6190, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics.

ZHAO Y., ZHANG H., ZHOU S. & ZHANG Z. (2020). Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 1796–1806, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.162.

ZHDANOV F. (2019). Diverse mini-batch active learning. *CoRR*, **abs/1901.05954**.

ZHOU B., CAI X., ZHANG Y., GUO W. & YUAN X. (2021). Mtaal : Multi-task adversarial active learning for medical named entity recognition and normalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(16), 14586–14593. DOI : 10.1609/aaai.v35i16.17714.

ZHU J. & HOVY E. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, p. 783–790, Prague, Czech Republic : Association for Computational Linguistics.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Détecter une erreur dans les phrases coordonnées au sein des rédactions universitaires

Laura Noreskal[1] Iris Eshkol-Taravella[1] Marianne Desmets[2]
(1) MoDyCo UMR 7114, 200 avenue de la République, 92000 Nanterre, France
(2) LLF UMR 7110, 5, rue Thomas Mann, 75205 Paris cedex 13
Laura.noreskal@parisnanterre.fr, ieshkolt@parisnanterre.fr,
marianne.desmets@parisnanterre.fr

RÉSUMÉ

Beaucoup d'étudiants rencontrent des difficultés dans la maîtrise du français écrit. Sur la base d'une enquête linguistique préliminaire, il est apparu que les constructions syntaxiques comprenant des coordinations et des constructions elliptiques forment des contextes linguistiques sensibles aux erreurs ou aux maladresses dans les écrits des étudiants. Dans le cadre du projet écri+, notre recherche vise à développer un outil de détection automatique de phrases coordonnées erronées dans les rédactions des étudiants afin de leur permettre de s'auto-former en expression écrite. Après avoir constitué le corpus de phrases coordonnées extraites des différents écrits universitaires (exercices, examens, devoirs, rapports de stage et mémoires), nous avons établi une typologie des erreurs qui a servi de modèle pour l'annotation du corpus. Nous avons entrainé premièrement des classifieurs (Random Forest, SVM, CamemBERT et FlauBERT) afin de détecter deux étiquettes: erronée et correcte puis, dans un second temps, un classifieur multi-label pour diagnostiquer l'erreur.

ABSTRACT

Detecting error in coordinated sentences in students' writings.
Many students have difficulties in mastering written French. Based on a preliminary linguistic investigation, it appeared that syntactic constructions, including coordinations and elliptical constructions, are linguistic contexts sensitive to errors or awkwardness in students' writing. As part of écri+ project, the aim of our research is to develop a tool for automatic detection of erroneous coordinated sentences in students' essays in order to enable them to train themselves in written expression. After having constituted the corpus of coordinated sentences extracted from different academic writings (exercises, exams, homework, internship reports and dissertations), we established a typology of errors which served as a model for the annotation of the corpus. We first trained classifiers (Random Forest, SVM, CamemBERT and FlauBERT) to detect two labels: erroneous and correct, and then, in a second step, a multi-label classifier to diagnose the error.

MOTS-CLES : écri+, erreurs syntaxiques, phrases coordonnées, typologie d'erreurs, apprentissage automatique de surface, apprentissage profond, rédaction des étudiants.

KEYWORDS: écri+, syntactic errors, corpus, coordinated sentences, errors typology, machine learning, deep learning, student writings

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 248–261.

1 Contexte de la recherche

Beaucoup d'étudiants rencontrent des difficultés dans la maîtrise du français écrit. Face à cette situation un réseau d'universités mobilisant une centaine d'enseignants et de chercheurs s'est associé au projet écri+[1] pour développer des méthodes pédagogiques innovantes. En se basant sur des recherches en informatique, en didactique et en linguistique, le projet vise à proposer une solution nationale pour accompagner, former, évaluer et certifier les étudiants du premier cycle universitaire. Sur la base d'une enquête linguistique préliminaire, il est apparu que les constructions syntaxiques comprenant des coordinations et des constructions elliptiques forment des contextes linguistiques sensibles aux erreurs ou aux maladresses dans les écrits des étudiants (section 3.1). Notre tâche dans ce projet est de développer un outil capable de détecter automatiquement les erreurs syntaxiques intra-phrastiques dans les constructions coordonnées afin de permettre aux étudiants de s'auto-former en expression écrite. Pour repérer les constructions coordonnées dans les rédactions des étudiants, nous nous sommes basées sur les travaux de Martinet (1980), de Goosse et al. (2008), de Riegel et al. (2009), d'Abeillé et Godard (2021) qui distinguent deux types de coordination : la coordination explicite qui requiert un coordonnant et la coordination implicite qui n'en attend pas. De plus, le rôle des éléments conjoints est important car il doit être respecté à chaque ajout. Nous avons décidé d'étudier les structures qui contiennent de la coordination explicite, soit des conjonctions de coordination (*mais, ou, et, or, ni, car, soit. . . soit*) ou des adverbes de liaison (*puis, ensuite, cependant, néanmoins*. . .).

2 Constitution des corpus

Dans l'optique d'observer les constructions coordonnées et de relever les différents types d'erreurs récurrentes dans les rédactions des étudiants, il était nécessaire de constituer un corpus. Plusieurs méthodes étaient alors envisageables : (1) un *corpus contrôlé* (Tellier, 2012) nécessitant de préparer un protocole expérimental qui inciterait les étudiants à produire des phrases coordonnées ; (2) un *corpus semi-contrôlé*, c'est-à-dire un ensemble de données langagières produites lors d'une formation comme les mémoires ou les devoirs maison ; (3) le recueil d'un corpus écologique constitué de productions naturelles des étudiants telles que les notes de cours. Parmi ces trois types de corpus, la deuxième solution nous a semblé plus pertinente. En effet, la mise en place d'un protocole expérimental demande de proposer une consigne qui assure d'avoir des phrases coordonnées, une consigne pas assez claire ou trop exigeante pourrait donner des résultats inutilisables. Quant au corpus écologique, il est difficile de savoir si les étudiants utilisent beaucoup de structures coordonnées dans leurs prises de notes. Il nous a donc semblé plus pertinent de collecter un corpus semi-contrôlé constitué de productions réalisées dans le cadre de la formation des étudiants qui selon nous contiendraient davantage de structures coordonnées que les productions naturelles.

2.1 Corpus de rédactions

Nous avons collecté les productions dites « évaluatives », c'est-à-dire les productions réalisées dans le but d'être évaluées par un enseignant dans le cadre d'un enseignement supérieur. Parmi ces productions, nous avons retenu 4 types de rédactions : 139 devoirs maison, 167 exercices faits en classe, 47 rapports de stage et 27 mémoires, ce qui correspond au total à 380 rédactions. Les écrits proviennent de différents niveaux d'étude allant de la première année de licence à la deuxième année de master. Les domaines disciplinaires concernés sont les sciences du langage, l'histoire et le droit.

[1] anr-17-NCUN-00015

2.2 Corpus de phrases coordonnées

À partir du corpus de rédactions, nous avons pu extraire les structures coordonnées en utilisant des patrons morphosyntaxiques créés avec l'outil Unitex (Paumier, 2011) qui reconnaissent les conjonctions de coordination et les adverbes de liaison. Sur un total de 39 692 phrases, 6 645 phrases contenant de la coordination ont été extraites (voir tableau 1).

Types	**Nombre de rédactions**	**Total des phrases**	**Phrases avec coordination**
Devoirs maison	139	7 635	1 467
Exercices	167	2 502	593
Mémoire	27	13 244	1890
Rapports	47	16 311	2695
Total	380	39 692	6645

TABLEAU 1 : Composition du corpus de phrases coordonnées

3 Annotation manuelle

Le processus d'annotation a été mis en place par 3 annotateurs (une experte et deux linguistes non-experts). Cette annotation répond à deux objectifs : (1) proposer une analyse quantitative des erreurs dans les rédactions étudiantes ; (2) créer un corpus de référence pour l'apprentissage automatique de ces erreurs. L'annotation consistait à renseigner sur la présence ou non d'une erreur dans une structure coordonnées et sur son type. Nous avons annoté d'abord une partie des phrases coordonnées avec les étiquettes **correcte** et **erronée**. À partir de là, nous avons pu établir une typologie des erreurs et nous concentrer sur une annotation en types d'erreur.

3.1 Typologie des erreurs

Après avoir observé les phrases erronées collectées, nous avons pu distinguer 8 types d'erreurs que nous avons regroupés en 5 catégories : prépositions, syntagmes conjoints, accords sujet-verbe distant, ponctuation et autres.

3.1.1 Les prépositions

Les prépositions sont souvent sujettes aux erreurs dans les constructions coordonnées. Trois sous-types d'erreurs sont distingués : l'ajout d'une préposition non-attendue (PREP ADD) (1), le remplacement d'une préposition par une autre (PRED REMP) (2) et l'absence de préposition (PREP ABS) (3).

(1) *Avant de visionner la comédie musicale, il faudra étudier avec les élèves la période révolutionnaire pour comprendre les raisons de la Révolution et **de** rendre cette activité ludique mais pédagogique.

Dans cet exemple, le problème vient de la préposition non-attendue *de* qui forme un syntagme prépositionnel avec le syntagme verbal *rendre cette activité ludique mais pédagogique*.

(2) *Le fait de les aider à se construire et **à** les voir grandir, tout en leur apportant un savoir doit être gratifiant et réjouissant.

L'exemple (2) est considéré comme erroné car le syntagme prépositionnel *à les voir grandir* est introduit par la mauvaise préposition (*à* au lieu de *de*). En effet, le syntagme *à les voir grandir* est considéré comme le complément du verbe *aider* or il s'agit en réalité du complément de *fait* qui doit être introduit par *de*.

(3) *Ils illustrent leur propos en appliquant cette analyse aux appendices et les formules illocutoires, les actes indirects et les questions biaisées.

En (3), le problème est lié aux compléments du verbe *appliquer*, lequel attend généralement un complément direct mais peut prendre également un complément indirect introduit par la préposition *à*. Dans la phrase (3), le premier complément indirect *aux appendices* respecte la valence du verbe mais pas les autres compléments.

3.1.2 Les syntagmes conjoints

— Mauvaise cohérence des groupes syntaxiques (MCGS)

(4) *Par ailleurs, il appartient à tous les personnels de transmettre aux élèves les valeurs et doivent avoir un devoir de neutralité.

Dans l'exemple (4), les deux conjoints n'ont pas le même rôle syntaxique ni la même forme. De plus, il n'est pas possible de retrouver le sujet du syntagme verbal *doivent avoir un devoir de neutralité* qui représente le second conjoint.

— Grande distance entre conjoints (DIST CONJOINT)

(5) *S'associer avec des associations telles que celles du Téléthon qui permet de sensibiliser les élèves, de transmettre des valeurs républicaines, ainsi que Nettoyons la Nature.

Dans l'exemple (5), les deux conjoints (*Téléthon* et *Nettoyons la Nature*) reliés par *ainsi que* sont trop éloignés l'un de l'autre. Cela peut produire une incompréhension chez le lecteur.

3.1.3 Les accords entre sujet et verbe distant

(6) *Le personnage de droite est assis sur un tabouret et as une corpulence fine.

L'exemple (6) contient une erreurs d'accord entre le sujet et le verbe : le verbe avoir est conjugué à la deuxième personne du singulier alors que le sujet est à la troisième personne du singulier.

3.1.4 La ponctuation

Les erreurs de ponctuation sont également très fréquentes dans les phrases du corpus. Deux sous types d'erreurs sont proposés : les structures lourdes (7) et les problèmes d'absence de ponctuation (8).

— Structure Lourde (SL)

(7) *L'auteur dénonce les contrats sociaux très importants entre les pays du nord et les pays du sud plus précisément, la faim à l` échelle mondiale et il va donc accentuer ses disparités à travers une caricature qui à une visée humoristique mais également critique puisque elle met en relief les pays du Nord , riches et industrialisés, dominé par la Triade qui regroupe les grands pôles économiques du monde et les pays du Sud ,

pauvres et désigné comme le « Tiers-monde »qui réfléchissent à une solution pour lutter contre la famine autour d'un repas .

L'exemple (7) est considéré comme erroné car il comprend de nombreuses propositions imbriquées les unes dans les autres, ce qui rend la compréhension difficile.

— Absence de ponctuation (PONC ABS)

(8) *Chaque année des milliers de sacs à main tous différents les uns des autres rentrent sur le marché mondial et ces sacs à main répondent à une demande croissante de la part des femmes donc je me dit que c'est vraiment un effet de mode dont certaines femmes ne pourraient plus se passer.

Dans l'exemple (8), il n'y a aucune autre ponctuation dans le texte, à part le point final. Plusieurs segments s'enchainent avec les conjonctions de coordination comme seuls liens. Cela peut alors demander plus d'efforts cognitivement pour la lecture et la compréhension.

3.1.5 Autres

La classe Autres comprend les erreurs qui ne font pas partie des classes précédentes. Les phrases réunies dans Autres contiennent des erreurs différentes mais peu fréquentes comme la présence de *pas* avant *ni* (9).

(9) Pour ma part, je ne connaissais pas ni le master français langue étrangère (FLE) ni le métier du traitement automatique des langues.

3.2 Accord inter-annotateurs

Après avoir développé un guide d'annotation basé sur la typologie ci-dessus, nous avons lancé une campagne d'annotation et avons calculé un accord inter-annotateurs entre deux annotateurs non-experts. Pour vérifier la cohérence de la typologie, une experte a formé les non-experts sur la reconnaissance des différents types erreurs. À la suite de la formation, les deux annotateurs ont annoté la présence ou non d'erreurs et le type d'erreurs dans 200 phrases. L'accord de la présence ou non d'une erreur a été calculé avec le kappa de Cohen et le score a atteint 0,88, ce qui est un accord excellent selon Landis & Coch (1977). L'accord inter-annotateur pour le type d'erreurs s'est élevé à 0,72 avec le kappa de Cohen. Les types d'erreurs les mieux reconnus ont été *DIST CONJ* et *PRED REMP*. Concernant les erreurs *PRED REMP*, lors de la lecture d'un texte, il se peut que le non-respect de la sous-catégorisation des verbes, noms ou adjectifs interpelle le lecteur qui repère ainsi qu'il s'agit de la mauvaise préposition. Quant à l'erreur *DIST CONJ*, elle touche directement à la compréhension de la coordination. Le lecteur a du mal à identifier les conjoints ce qui rend l'erreur plus facilement repérable. Une fois la typologie validée, 3145 phrases ont été annotées dont 1153 phrases erronées. La répartition des erreurs dans les phrases annotées est présentée dans le TABLEAU 2.

Types d'erreurs	Nombres
Préposition ajoutée (PREP ADD)	29
Préposition remplacée (PREP REMP)	44
Préposition absente (PREP ABS)	139
Mauvaise cohérence entre les groupes syntaxiques (MCGS)	35
Distance entre conjoint (DIST CONJ)	25
Mauvais accord sujet-verbe (MASV)	33

Structure lourde (SL)	244
Ponctuation absente (PONC ABS)	155
Autres	449
Total	1153

TABLEAU 2 : Constitution du corpus annoté

4 Détection automatique

De nombreuses recherches ont été faites sur la détection automatique des erreurs. Depuis quelques années la détection et la correction automatiques d'erreurs grammaticales (Grammatical Errors Detection, GEC en anglais) ainsi que les outils d'aide à la rédaction reçoivent beaucoup d'attention. Majoritairement développés dans le but d'aider les apprenants allophones (Garnier, 2014), les modèles de détection d'erreurs permettent souvent de reconnaître les erreurs d'accords (Fay-Varnier, 1990 ; Souque, 2014) ou les erreurs lexicales (Yuan *et al*., 2019). La détection d'erreurs orthographiques en français n'est pas en reste : Cordial de Synapse (1995), Antidote de Druide Informatique (1996) et ProLexis des Editions Diagonal (1997). Ces trois outils dominent le marché francophone qui, depuis quelques années, ne voit que très peu de nouvelles alternatives. De plus, les travaux se basant sur la syntaxe sont peu nombreux (Clément *et al.*, 2009). En anglais, trois types d'outils d'aide à l'écriture existent (Jourdan *et al*., 2023) : les outils de révision de phrases (Ouyang *et al*., 2022), les correcteurs grammaticaux (Tsai *et al.*, 2020) et les outils d'annotation de structures rhétoriques tels que AcaWriter (Knight *et al*., 2020). En nous concentrons sur le français, nous proposons de contribuer à la recherche dans ce domaine.

4.1 Détection binaire : erronée / correcte

La première étape de la détection d'erreurs syntaxiques dans les phrases coordonnées porte sur la détection des étiquettes **correcte** et **erronée** en utilisant les méthodes de l'apprentissage automatique de surface avec les deux classifieurs SVM et Random Forest et l'apprentissage profond fondé sur les modèles français CamemBERT (Martin *et al*., 2019) et FlauBERT (Le et al., 2019) en tant que classifieurs en utilisant Simple Transformers[2]. Pour ces expériences, nous avons utilisé deux corpus d'apprentissage : un corpus équilibré (600 phrases correctes et 600 phrases erronées) et un corpus déséquilibré en faveur des phrases erronées (300 phrases correctes et 900 phrases erronées) afin de vérifier si le déséquilibre peut aider à mieux détecter les erreurs. Le corpus de test est composé de 400 phrases et contient autant de phrases correctes que de phrases erronées. La répartition des erreurs dans les différents corpus est présentée dans le tableau suivant :

Types d'erreurs	Corpus équilibré	Corpus déséquilibré
Autres	167	320
Autres + PREP ADD	2	4
Autres + SL + PONC ABS	0	3
DIST CONJ	12	16
DIST CONJ + Autres	2	0
DIST CONJ + PONC ABS	0	4
MASV	16	23
MASV + Autres	2	2
MASV + SL	4	2

[2] https://simpletransformers.ai/

MCGS	18	23
MCGS + Autres	2	2
MCGS + PREP REMP	4	4
PONC ABS	100	119
PONC ABS + Autres	0	15
PONC ABS + PREP ADD	0	2
PONC ABS + PREP REMP	0	2
PREP ABS	56	73
PREP ABS + Autres	2	2
PREP ABS + PONC ABS	2	4
PREP ABS + PREP ADD + SL	3	3
PREP ABS + SL	6	0
PREP ADD	13	15
PREP ADD + SL + Autres	3	3
PREP REMP	24	25
PREP REMP + Autres	0	4
SL	150	172
SL + Autre	10	32
SL + PREP ABS	0	4
SL + PREP ADD	0	2
SL + PREP REMP	2	5
Total	600	900

TABLEAU 3 : Constitution des corpus d'entrainement

4.1.1 Apprentissage de surface

Nous avons utilisé une méthode de classification supervisée en utilisant deux classifieurs : Support Vector Machine (SVM) et Random Forest car ils ont obtenu de bons scores lors de la classification de documents lors des campagnes DEFT (DEFT 2021 et DEFT 2022).

4.1.1.1 Prétraitement

Notre chaîne de prétraitement comprend l'étiquetage morphosyntaxique, la lemmatisation, l'analyse syntaxique en dépendances, le chunking et d'autres traits résultant d'une observation du corpus. Chaque phrase est représentée par un ensemble de 20 traits linguistiques regroupés en trois classes :

— les traits généraux souvent utilisés lors du prétraitement des textes :
- les tokens
- les lemmes obtenus avec Treetagger (Schmid, 1994)
- les parties du discours obtenus avec *Treetagger*
- les chunks détectés grâce à *TreeTagger*,
- les relations de dépendances détectés grâce à la bibliothèque Python *spaCy*
- les trigrammes

— les traits binaires :
- la présence d'un verbe transitif grâce au dictionnaire électronique des mots (DEM) de Dubois (Dubois et al., 2010): en observant les phrases erronées, il est apparu que la sous-catégorisation des verbes transitifs n'était pas respectée. Nous cherchons donc à savoir si la seule présence d'un verbe transitif peut jouer un rôle dans l'apparition des erreurs.
- la présence d'une préposition : nous avons repéré plusieurs problèmes liés aux prépositions tels que *PREP ADD* ou *PRED REMP*. Face à cela, nous nous

sommes demandé si la présence d'une préposition pouvait être reliée à la présence d'une erreur.

- la présence de *que* : Lors de notre observation du corpus, nous avons pu remarquer que beaucoup de phrases erronées contenaient des propositions introduites par *que*. Nous avons donc ajouté ce trait afin d'observer si l'utilisation de *que* peut être liée à la présence d'une erreur.

— les traits numériques :

- le nombre de mots : en ajoutant le nombre de mots, nous espérons trouver une corrélation entre la longueur de la phrase et la présence d'une erreur. Le but de cette observation n'est pas de prescrire les phrases longues mais plutôt de savoir si ce type de phrases est plus sujet aux erreurs.
- le nombre de verbes transitifs : après avoir observé une tendance au non-respect de la sous-catégorisation des verbes, nous avons pensé qu'il serait intéressant de tester l'hypothèse selon laquelle le nombre de verbes transitifs aurait un impact sur la présence d'erreurs.
- le nombre de conjonctions de coordination : ce trait permet de vérifier s'il y a une corrélation entre le nombre de coordonnants et la présence d'erreurs dans une phrase.
- le nombre de *que* : la présence de *que* dans une phrase sous-entend que celle-ci est une phrase complexe avec une proposition subordonnée. De ce fait, chaque *que* présent dans une phrase la complexifie. Ainsi, nous cherchons à savoir si la présence de cette complexité dans une coordination peut être corrélée avec la présence d'une erreur.
- le nombre de prépositions : l'ajout de ce trait prend en compte le fait que les prépositions posent problème dans les phrases coordonnées. Nous cherchons alors à savoir si le nombre de prépositions dans une phrase peut être relié à la présence d'une erreur.
- le nombre de *à*, *de*, *sur*, *pour*, *dans* : nous avons ajouté ces traits reportant le nombre de chaque préposition dans une phrase afin d'observer si certaines prépositions sont plus sujettes aux erreurs que d'autres.
- le nombre de *ainsi que* : la locution conjonctive *ainsi que* joue un rôle de coordonnant dans certains de ses emplois. Nous avons pensé qu'il serait intéressant d'observer cette locution afin de savoir si ses propriétés peuvent poser problème.

Afin de sélectionner les traits les plus pertinents pour l'apprentissage, nous avons utilisé un algorithme de sélection de traits : RFE (Recursive Features Elimination). Lors de son application en validation croisée avec 5 échantillons, l'algorithme supprimait un ou plusieurs traits non pertinents pour l'apprentissage à chaque échantillonnage. Les 11 traits suivants ont été sélectionnés : l'analyse en dépendance, le chunking, les lemmes, le nombre de conjonctions de coordination, le nombre de *de,* le nombre de mots, le nombre de prépositions, le nombre de verbes transitifs, les parties du discours, les tokens et les trigrammes.

4.1.1.2 Expériences et résultats

Lors des expériences, nous avons testé deux aspects : le type de données (corpus équilibré/corpus déséquilibré) et les algorithmes de classification (Random Forest/SVM). De plus, afin d'optimiser les performances de SVM et Random Forest, nous avons utilisé, lors de tous les tests, GridsearchCV, permettant de tester les différents paramètres d'un algorithme d'apprentissage pour en sélectionner les meilleurs.

Les expériences sur les deux corpus ont montré que les données déséquilibrées rendent la tâche de détection plus compliquée pour le classifieur. En effet, les résultats pour les données déséquilibrées sont mauvais avec des exactitudes de 0,5425 pour Random Forest et de 0,585 pour SVM. Quant aux résultats pour les données équilibrées, ils sont de 0,665 pour Random Forest et de 0,6325 pour SVM. Afin de mieux comprendre les résultats, nous avons observé les mesures de précision, rappel et f-mesure de chaque classe lors du test avec le corpus équilibré.

Random Forest	Précision	Rappel	F-mesure	Exactitude
Correcte	0,64	0,76	0,70	0,665
Erronée	0,70	0,57	0,63	

TABLEAU 4 : Random Forest : mesures de précision, rappel et f-mesure pour les deux classes

SVM	Précision	Rappel	F-mesure	Exactitude
Correcte	0,61	0,73	0,67	0,6325
Erronée	0,66	0,54	0,59	

TABLEAU 5 : SVM : mesures de précision, rappel et f-mesure pour les deux classes

Les résultats obtenus grâce à Random Forest montrent que la différence de f-mesure entre la classe correcte et la classe erronée est de 6%. Cependant, on remarque également que la précision pour la classe erronée (0.70) est plus élevée que celle de la classe Correcte (0.64). Cela signifie que le modèle est plus précis pour détecter les erreurs que pour détecter les phrases correctes. Néanmoins, le modèle rapporte moins de phrases erronées (0.57) que de phrases correctes (0.76).

4.1.2 Apprentissage profond

Pour l'apprentissage profond, nous avons utilisé deux modèles français pour une tâche de classification des phrases en **correcte**/**erronée** : CamemBERT et FlauBERT, avec *Simple Transformers*, une librairie d'*HuggingFace*. Les expériences ont été réalisées en variant le nombre d'époques.

	FlauBERT Corpus équilibré	FlauBERT Corpus déséquilibré	CamemBERT Corpus équilibré	CamemBERT Corpus déséquilibré
5 époques	0,5	**0,7**	**0,7125**	**0,77**
8 époques	0,71	0,5	0,695	0,73
10 époques	0,7	0,5825	0,685	0,72
15 époques	**0,715**	0,6275	0,6875	0,7375

TABLEAU 6 : Exactitudes des expériences menées en apprentissage profond

Nous observons que les résultats sont meilleurs que ceux que nous avons obtenus lors des expériences avec l'apprentissage de surface. Cela peut être dû au fait que les modèles utilisés pour l'apprentissage profond sont entraînés sur de gros corpus composés de phrases correctes, il pourrait être ainsi plus simple pour les modèles de détecter les phrases qui s'éloignent des phrases correctes apprises auparavant. Le meilleur résultat provient de l'expérience avec CamemBERT et le corpus déséquilibré lors des 5 époques (0.77).

	Précision	Rappel	F-mesure	Exactitude
Correcte	0,99	0,54	0,69	0,77

Erronée	0,68	0,99	0,81	

TABLEAU 7 : CamemBERT avec le corpus déséquilibré: mesures de précision, rappel et f-mesure pour les deux classes

Le modèle CamemBERT avec le corpus déséquilibré montre une très bonne précision (0,99) pour la classe correcte et un bon rappel (0,99) pour la classe erronée. La F-mesure de la classe erronée est aussi très élevée puisqu'elle atteint 0,81.

4.2 Détection multi-label

Un autre objectif de cette recherche est de réussir à diagnostiquer l'erreur en lui attribuant une étiquette qui détermine le type d'erreur mis en cause. Pour ce faire, nous avons utilisé la classification multi-label. Pour débuter la classification multi-label, nous avons gardé les données utilisées pour la classification binaire.

Lors de la classification multi-label, nous avons fait 4 expériences soit deux expériences avec CamemBERT et deux autres avec FlauBERT. Pour chaque modèle de langue, nous avons testé l'apprentissage avec un corpus déséquilibré et l'apprentissage avec un corpus équilibré. Le nombre d'époque a été figé à 5 afin de ne pas avoir de dégradation des résultats comme pour la classification binaire.

Modèles	Corpus	Bien classées	Mal classées	Exactitude
CamemBERT	Equilibré	18	182	4,5
	Déséquilibré	218	182	54,5
FlauBERT	Equilibré	83	317	20,75
	Déséquilibré	176	224	44

TABLEAU 8 : Exactitudes des expériences menées pour la détection multi-label

En observant les résultats nous remarquons que seul le modèle CamemBERT avec le corpus déséquilibré a une exactitude supérieure à 50%. En somme, les résultats sont assez mauvais puisque les modèles peinent à classer correctement les phrases. Pour mieux comprendre ces erreurs, nous avons observé les résultats des différentes classifications.

De façon générale, les modèles ont tendance à combiner l'étiquette *None,* qui correspond à l'absence d'erreur, à une autre étiquette d'erreur telles que *SL* et autres. Seul le modèle CamemBERT avec le corpus déséquilibré ne reproduit pas cette erreur. Quant au modèle CamemBERT avec le corpus équilibré, il est celui qui est le plus touché par cette erreur d'étiquetage puisqu'il comptabilise 200 phrases annotées *None* et *SL*. Les modèles FlauBERT sont aussi touchés par cette erreur. FlauBERT avec le corpus équilibré produit cette erreur sur 148 phrases alors que FlauBERT avec le corpus déséquilibré la produit sur 56 phrases.

Concernant les phrases bien classées, elles concernent en général les types d'erreurs *PREP ADD*, *PREP ABS* et MASV (*mauvais accord Sujet-Verbe*). Aussi, les phrases correctes sont bien classées par les modèles entrainés avec les corpus déséquilibrés alors que les autres modèles arrivent surtout à annoter les phrases avec les erreurs d'*absence de préposition*.

	VP	VN	FP	FN	Précision	Rappel	F-mesure
Autres	0	270	3	127	0	0	0

DIST CONJ	0	396	1	3	0	0	0
MASV	3	391	6	0	0,33	1	0,5
MCGS	0	396	1	3	0	0	0
PONC ABS	20	380	0	0	1	1	1
PREP ABS	11	389	0	0	1	1	1
PREP ADD	4	396	0	0	1	1	1
PREP REMP	3	396	0	1	0,75	1	0,86
SL	127	249	0	24	1	0,84	0,91

TABLEAU 9 : Mesures de la détection multilabel avec CamemBERT et le corpus déséquilibré

En observant les mesures sur ce modèle, on réalise que certains types d'erreurs sont mieux reconnus que d'autres. Les résultats montrent que ce modèle reconnaît très bien les erreurs PONC ABS, PREP ABS et PREP ADD mais qu'il est incapable de détecter les erreurs DIST CONJ, Autres et MCGS. Les modèles d'apprentissage profond n'indiquant pas réellement ce qui est pris en compte pour la détection, il est difficile de préciser la raison pour laquelle certaines erreurs seront mieux reconnues que d'autres. Cependant, il semblerait que les phrases correctes soient mieux reconnues lors des entrainements avec les corpus déséquilibrés car les erreurs étant plus nombreuses dans le corpus, il devient plus simple pour l'algorithme de différencier une erreur d'une phrase correcte. Pour les *PREP ABS*, la raison pour laquelle ils sont bien repérés par tous les modèles pourrait être due au fait qu'il y a plus de données que pour d'autres erreurs. Une augmentation du corpus serait alors bénéfique pour tous les autres types d'erreurs.

En résumé, la détection multi-label n'a pas donné de résultats probants puisque nous avons obtenu des exactitudes très faibles et des erreurs d'étiquetage combinant l'étiquette *None* à d'autres étiquettes d'erreurs. En utilisant l'apprentissage profond, il est difficile de savoir ce qui permet de classer une phrase dans un type d'erreur plutôt qu'un autre, ainsi il serait intéressant d'envisager une approche symbolique pour cibler et mieux localiser les erreurs dans les phrases. En testant ce type de classification automatique, nous avons cependant pu observer que certains types d'erreurs sont mieux reconnus que d'autres.

5 Conclusion et perspectives

Cette recherche vise à détecter automatiquement les phrases coordonnées erronées dans les rédactions des étudiants. Après avoir constitué le corpus de phrases coordonnées extraites des différents écrits universitaires (exercices, examens, devoirs, rapports de stage et mémoires), nous avons établi une typologie des erreurs que nous avons validée par un accord inter-annotateurs. Par la suite, nous avons procédé à plusieurs expériences portant sur l'apprentissage profond et l'apprentissage de surface. Les premières expériences concernaient la détection des étiquettes **erronée** et **correcte**. L'apprentissage profond a donné de meilleurs résultats puisque nous avons obtenu une exactitude de 0,77 avec le modèle CamemBERT. Puis, nous avons testé la détection multi-label en utilisant l'apprentissage profond. Nos résultats n'ont pas été probants mais ils laissent à penser qu'il faudrait sûrement observer plus précisément les résultats et peut-être aussi envisager d'autres approches comme des méthodes symboliques qui permettraient de mieux cibler les erreurs. Il serait également utile de revoir la typologie des erreurs afin de voir si les types d'erreurs les moins bien reconnus comme structures lourdes (SL) et absence de ponctuation (PREP ABS) sont vraiment pertinents pour notre recherche. En effet, lors des apprentissages, ce sont les types d'erreurs qui ont été les moins bien reconnus. Ces deux types se fondent sur l'utilisation de la ponctuation, ainsi, il se pourrait que les règles adoptées pour catégoriser les erreurs de ponctuation ne soient pas assez distinctives pour que ces erreurs soient repérables par les algorithmes.

Références

Abeillé, A., Godard, D., en collab. Avec Delaveau A. & Gautier A. (2021). *La Grande Grammaire du français*. Actes Sud / Imprimerie Nationale.

Clément, L., Gerdes, K. et Marlet, R. (2009). Grammaires d'erreur–correction grammaticale avec analyse profonde et proposition de corrections minimales In Nazarenko, A. & Poibeau, T. Éds., *Actes de TALN 2009 (Traitement automatique des langues naturelles),* p. 158–167, Senlis, France.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20.1, p. 37-46.

Dubois, J. et Dubois-Charlier, F. (2010). La combinatoire lexico-syntaxique dans le Dictionnaire électronique des mots. Les termes du domaine de la musique à titre d'illustration. *Langages* 3, p. 31-56.

Fay-Varnier, C. (1990). *Aide à la détection de fautes grammaticales par une analyse progressive des phrases*. Thèse de doctorat. Institut National Polytechnique de Lorraine.

Garnier, M. (2014). *Utilisation de méthodes linguistiques pour la détection et la correction automatisées d'erreurs produites par des francophones écrivant en anglais*. Thèse de doctorat. Université Toulouse le Mirail-Toulouse II.

Goosse, A. et Grevisse, M. (2008). *Le bon usage*. De Boeck Superieur. Louvain-la-Neuve.

Grouin, C. et Illouz, G. (2022). Notation automatique de réponses courtes d'étudiants : présentation de la campagne DEFT 2022 (Automatic grading of students' short answers : presentation of the DEFT 2022 challenge). In Y. Estève, T. Jiménez, T. Parcollet, M. Zanon Boito, Éds., *Actes de TALN 2022 (Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*, p. 1-10, Avignon, France.

Grouin, C., Grabar, N. et Illouz, G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne DEFT 2021. In *Actes de TALN 2021 (Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*, p. 1-13, Lille, France.

Jourdan, L., Boudin, F., Dufour, R., Hernandez, N. (2023). Text revision in Scientific Writing Assistance: An Overview. In *13th International Workshop on Bibliometric-enhanced Information Retrieval (BIR 2023)*, Dublin (IE), Ireland. ⟨hal-04053934⟩.

Knight, S., Shibani, A., Abel, S., Gibson, A., Ryan, P. (2020) Acawriter: A learning analytics tool for formative feedback on academic writing. *Journal of Writing Research.*

Landis, J. R. et Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, p. 363-374.

Le, H., Vial, L., Frej, F., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L. et Schwab, D. (2020). FlauBERT: Unsupervised Language Model Pre-training for

French. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 2479–2490, Marseille, France. European Language Resources Association.

Li, X., Michel, P., Anastasopoulos, A., Belinkov, Y., Durrani, N., Firat, O., Koehn, P., Neubig, G., Pino, J. et Sajjad, H. (2019). Findings of the First Shared Task on Machine Translation Robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1),* p. 91–102, Florence, Italy. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. et Stoyanov, V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. arXiv abs/1907.11692. Retrieved from https://arxiv.org/abs/1907.11692.

Noreskal, L., Eshkol-Taravella, I. & Desmets, M. (2021). Erroneous Coordinated Sentences Detection in French Students' Writings. *Communications in Computer and Information Science 1463*, p. 586-596.

Martin, L., Müller, B., Suárez, P. J. O., Dupont, Y., Romary, L., Villemonte de la Clergerie, E. et Seddah, D. et Sagot, B. (2019). « CamemBERT: a Tasty French Language Model». arXiv: 1911.03894.

Martinet, André. (1980). *Éléments de linguistique générale*. Collection U Prisme. Albin Michel.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022) Training language models to follow instructions with human feedback, arXiv preprint arXiv:2203.02155.

Suárez, O., Javier, P., Romary, L. et Sagot, B. (2020). A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. p. 1703-1714. Association for Computational Linguistics.

Riegel, M., Pellat, J-C. et Rioul, R. (2008). *Grammaire méthodique du français*. Édition PUF. Paris.

Schmid, H. (1994). TreeTagger-a language independent part-of-speech tagger. Institut Für Maschinelle Sprachverarbeitung: Universität Stuttgart.

Souque, A. (2014). *Modèle de vérification grammaticale automatique gauche-droite*. Thèse de doctorat. Université de Grenoble.

Tellier, M. (2012). De l'usage du corpus semi-contrôlé dans la recherche en didactique des langues. *Rencontres de l'ASDIFLE. FLE : L'instant et l'histoire 49 et 50*. Clé International, p. 39-47.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS ». *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), p. 2214-2218.

Tsai, C.-T., Chen, J.-J., Yang, C.-Y., Chang, J. S. (2020) LinggleWrite: a coaching system for essay writing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, p. 127–133, Association for Computational Linguistics.

Wenzek, G., Lachaux, M-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A. et Grave, E. (2020). CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. In Proceedings of the Twelfth Language Resources and Evaluation Conference, p. 4003–4012, Marseille, France. European Language Resources Association.

Yuan, Zheng, Felix Stahlberg, Marek Rei, Bill Byrne et Helen Yannakoudakis (2019). Neural and FST-based approaches to grammatical error correction. *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, p. 228-239.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Production automatique de gloses interlinéaires à travers un modèle probabiliste exploitant des alignements

Shu Okabe François Yvon
Université Paris-Saclay, CNRS, LISN, Bât. 508, Rue du Belvédère, F-91405 Orsay, France
{shu.okabe,francois.yvon}@limsi.fr

RÉSUMÉ

La production d'annotations linguistiques ou *gloses interlinéaires* explicitant le sens ou la fonction de chaque unité repérée dans un enregistrement source (ou dans sa transcription) est une étape importante du processus de documentation des langues. Ces gloses exigent une très grande expertise de la langue documentée et un travail d'annotation fastidieux. Notre étude s'intéresse à l'automatisation partielle de ce processus. Il s'appuie sur la partition des gloses en deux types : les gloses grammaticales exprimant une fonction grammaticale et les gloses lexicales indiquant les unités de sens. Notre approche repose sur l'hypothèse d'un alignement entre les gloses lexicales et une traduction ainsi que l'utilisation de Lost, un modèle probabiliste de traduction automatique. Nos expériences sur une langue en cours de documentation, le tsez, montrent que cet apprentissage est effectif même avec un faible nombre de phrases de supervision.

ABSTRACT

A Probabilistic Model for Automatic Interlinear Glossing and Alignment.

An important task in language documentation is the generation of linguistic annotations, or *interlinear glosses*), which express the meaning or role of each unit identified in a spoken source utterance (or its transcription). Glossing requires extensive expertise in the studied language and tedious work. In this article, we study ways to automate this process. Two types of glosses exist : grammatical glosses expressing a grammatical function and lexical glosses carrying the meaning of the unit. Our approach assumes that lexical glosses can be aligned with a target translation, enabling us to repurpose Lost, a probabilistic translation model for the glossing task. Our experiments on Tsez, a language in the process of being documented, show that useful glosses can be learned, even with a small number of supervision sentences.

MOTS-CLÉS : génération de gloses interlinéaires, documentation automatique des langues, alignement de mots.

KEYWORDS: interlinear gloss generation, computational language documentation, word alignment.

1 Introduction

Dans leur travail de documentation des langues, les linguistes de terrain produisent différentes strates d'annotations linguistiques des données orales collectées. Ces annotations permettent d'étudier plus en détail la langue et de préparer la production de dictionnaires ou de grammaires.

La figure 1 illustre ces différentes strates d'annotations appliquées à la transcription phonétique

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 262–274.

S [source] :	sidaquł	yiła–r	ciq–q	allah–s	ašuni	b–ukad–n
G [gloses] :	one.day	DEM2.IISG.OBL–LAT	forest–POSS.ESS	God–GEN1	belt	III–see–PST.UNW
G' [catégories] :	LEX	GRAM–GRAM	LEX–GRAM	LEX–GRAM	LEX	GRAM–LEX–GRAM
T [traduction] :	*one day she saw a rainbow in the forest*					

FIGURE 1 – Exemple de strates d'annotation linguistique dans le cadre de la documentation de langue.

de l'enregistrement d'une phrase isolée. La phrase **S** dans la langue source étudiée (ici le tsez) est segmentée en deux niveaux : les unités lexicales sont séparées par des espaces et les morphèmes sont indiqués par les tirets au sein d'un mot. Une seconde étape d'annotation consiste à renseigner le sens ou la fonction grammaticale de chaque morphème. Ce niveau d'annotation est appelé *glose interlinéaire* (**G** sur la figure 1). Nous distinguons deux catégories de gloses sur la ligne **G'** : les gloses *grammaticales*, telles que LAT, indiquent la fonction du morphème ; les gloses *lexicales* (comme forest) expriment le sens du morphème, en utilisant un concept dans une langue de documentation (ici l'anglais). Enfin, une traduction **T** accompagne chaque phrase.

Si les phrases sources et leurs traductions peuvent être recueillies simultanément sur le terrain, elles ne sont glosées qu'ultérieurement lors d'étapes d'analyse. Elles sont ainsi coûteuses à obtenir et exigent une grande expertise et un fastidieux travail d'annotation manuelle. Il n'est alors pas surprenant d'observer que les ressources complètement annotées soient peu nombreuses au regard du volume d'enregistrements bruts (Seifart *et al.*, 2018). Notre objectif est ici d'étudier comment une partie de ce processus pourrait être automatisé, en effectuant une pré-annotation qui serait ensuite révisée par des annotateurs. En effet, les phénomènes complexes, qui, souvent, intéressent davantage les linguistes, ne sont pas les plus fréquents ; une automatisation pourrait être bénéfique pour traiter les cas les plus courants. Elle permettrait également d'améliorer la cohérence et la vitesse de l'annotation en gloses (Baldridge & Palmer, 2009). La tâche que nous considérons consiste alors à calculer des gloses (**G**) à partir de la phrase source segmentée en morphèmes (**S**) et de sa traduction (**T**).

Cette tâche soulève un grand nombre de difficultés, comme la faible quantité de phrases disponibles pour superviser cette annotation. De plus, si la variété de gloses grammaticales est en nombre fini (s'apparentant alors à une tâche d'étiquetage classique), les gloses lexicales sont, par nature, en nombre quasi illimité. Enfin, malgré certaines conventions partagées (comme les *Leipzig Glossing Rules* [1] (Bickel *et al.*, 2008)), ces annotations s'appuient sur une interprétation linguistique, toujours sujette à des variations inter- et intra-personnelles. Face à ces constats, la production automatique de gloses pour les langues en cours de documentation a été abordée de différentes manières, principalement sous l'angle d'un étiquetage de séquences impliquant plusieurs étapes (Samardžić *et al.*, 2015; Moeller & Hulden, 2018; Barriga Martínez *et al.*, 2021). Une approche classique repose sur l'utilisation de champs markoviens conditionnels (*Conditional Random Field*, CRF) (Lafferty *et al.*, 2001; Tellier & Tommasi, 2011). Par exemple, (McMillan-Major, 2020) emploie deux CRF : l'un pour prédire les gloses depuis la phrase source, un autre depuis la phrase traduite. Les deux prédictions sont ensuite combinées afin d'obtenir la prédiction finale. (Zhao *et al.*, 2020) met en œuvre une approche neuronale, considérant les deux entrées (phrases source et cible) comme séparées dans leur architecture. Ces approches illustrent les diverses solutions imaginées pour aborder le principal défi de la tâche de génération de gloses : le problème posé par l'inventaire des gloses lexicales.

Pour y répondre, nous supposerons qu'il est possible de dériver les gloses lexicales de la traduction.

1. Un inventaire en français des principales étiquettes grammaticales est proposé par B. Fradin (voir http://www.llf.cnrs.fr/fr/node/60).

Cette hypothèse permet de circonscrire l'ensemble des étiquetages d'une phrase donnée lors de l'entraînement et de l'inférence d'un modèle probabiliste, qui pourra alors prendre en charge un ensemble arbitraire d'étiquettes. Dans la section 2, après avoir formalisé la tâche, nous discutons du calcul de ces alignements, puis de leur considération dans un modèle de glose automatique. La section 3 présente les résultats obtenus sur le tsez, une langue déjà utilisée dans (Zhao *et al.*, 2020) [2].

2 Un modèle probabiliste pour les gloses interlinéaires

2.1 Formalisation du problème

Considérons tout d'abord la ligne **G'** de la figure 1. Son calcul peut se formaliser comme une tâche d'étiquetage de séquence classique, où chaque morphème source est associé à une étiquette binaire : LEX ou GRAM. Pour ce faire, une méthode standard est d'utiliser un CRF qui modélise :

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp\left\{\sum_{k=1}^{K}\theta_k G_k(\boldsymbol{x},\boldsymbol{y})\right\}}{\sum_{\boldsymbol{y'}\in\mathcal{Y}^T}\exp\left\{\sum_{k=1}^{K}\theta_k G_k(\boldsymbol{x},\boldsymbol{y'})\right\}} = \frac{1}{Z_\theta(\boldsymbol{x})}\exp\left\{\sum_{k=1}^{K}\theta_k G_k(\boldsymbol{x},\boldsymbol{y})\right\}, \quad (1)$$

où $\boldsymbol{x}$ représente la séquence de T morphèmes sources et $\boldsymbol{y}$ les étiquettes associées. Nous noterons $\mathcal{Y}$, l'ensemble des étiquettes possibles, ici restreint à deux possibilités $\mathcal{Y} = \{\text{LEX}, \text{GRAM}\}$. L'espace de recherche contenant l'ensemble des étiquetages possibles pour une phrase est alors égal à $\mathcal{Y}^T$. $\{G_k(\boldsymbol{x},\boldsymbol{y}), k \in [1,K]\}$ sont les caractéristiques et $\boldsymbol{\theta} \in \mathbb{R}^K$, le vecteur de paramètres associé. Une caractéristique teste des propriétés locales du couple $(\boldsymbol{x},\boldsymbol{y})$ à une position donnée (unigramme) ou à deux positions consécutives (bigramme). Enfin, $Z_\theta(\boldsymbol{x})$ est la fonction de partition qui normalise sur tous les étiquetages possibles et permet d'interpréter (1) comme une probabilité. L'apprentissage des paramètres à partir d'un corpus associant morphèmes et étiquettes est un processus standard et amplement documenté.

Il est possible d'étendre ce modèle pour prédire chaque glose *grammaticale* plutôt qu'une seule étiquette (GRAM). Ce changement implique un accroissement du nombre d'étiquettes possibles et de l'espace de recherche. Les calculs associés restent en effet réalisables même lorsque l'on considère plusieurs centaines d'étiquettes (Mueller *et al.*, 2013; Lavergne & Yvon, 2017). Cette méthode est utilisée dans des travaux précédents de prédiction de gloses par (Moeller & Hulden, 2018; Barriga Martínez *et al.*, 2021; Okabe & Yvon, 2022). Une fois les étiquettes grammaticales prédites par un CRF, les gloses lexicales (toutes regroupées sous l'étiquette LEX) sont annotées manuellement.

Calcul des gloses lexicales La prise en charge des gloses lexicales se heurte au problème de leur nombre et inventaire, non fixés au préalable. Deux hypothèses sont alors possibles :

H1 : considérer que seules les gloses lexicales observées à l'entraînement sont possibles, ce qui permet de spécifier *globalement* $\mathcal{Y}$, dont la taille pourra toutefois poser problème ;

2. Le code pour reproduire les expériences est disponible à l'adresse : https://github.com/shuokabe/gloss_lost.

H2 : supposer que les gloses lexicales peuvent également être déduites de la traduction, dans laquelle on peut s'attendre à retrouver les mêmes concepts évoqués. Cette hypothèse est notamment explorée par (McMillan-Major, 2020; Zhao *et al.*, 2020).

Sous l'hypothèse [H2], que nous adoptons également ici, gloser revient à prédire, pour chaque morphème, soit une étiquette grammaticale, soit un mot de la traduction (plus précisément, son lemme). [H1] et [H2] ne sont pas complètement exclusives et il est aussi possible d'inclure (certains) des mots observés à l'entraînement, même s'ils n'apparaissent pas dans la traduction. Ceci s'avère en particulier nécessaire, comme dans l'exemple 1, car les gloses lexicales comme God ou belt ne peuvent pas être déduites de la seule traduction (même en explorant les synonymes). L'utilisation d'un lexique complémentaire permet donc d'associer une glose lorsque celle-ci ne figure pas dans la traduction. La figure 2 illustre notre formalisation pour la phrase de l'exemple 1.

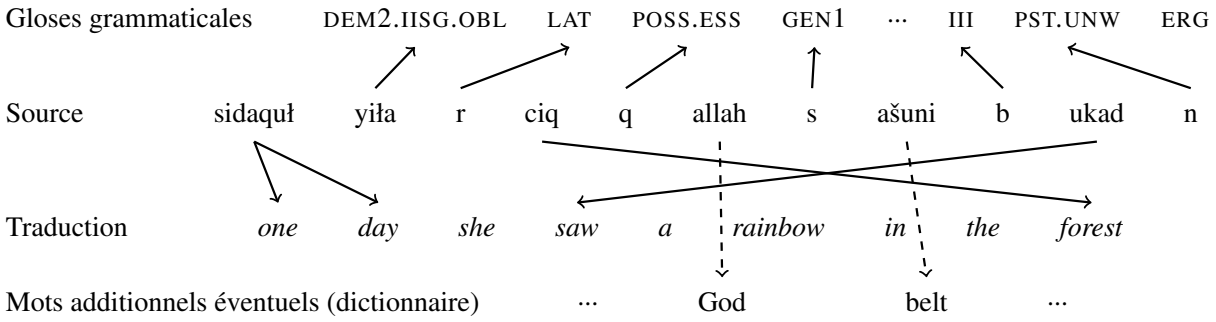


FIGURE 2 – Illustration du calcul de gloses pour la phrase de l'exemple 1. La partie supérieure contient les gloses grammaticales ; la partie inférieure, un alignement partiel entre la source et la traduction, ainsi que les mots supplémentaires (issus d'un dictionnaire).

Mettre en œuvre un modèle probabiliste inspiré de (1) demande de résoudre plusieurs problèmes : (a) définir les étiquetages possibles pour une phrase source donnée en exploitant les mots de la traduction et les gloses du corpus d'apprentissage, tout en prenant soin de contraindre l'espace de recherche associé pour que le calcul de Z_θ reste faisable ; (b) définir les étiquetages de référence pour superviser l'apprentissage, car les alignements entre source et traduction ne sont pas observés ; (c) celui de définir l'ensemble des étiquetages possibles à l'inférence.

Notre approche propose des réponses à chacun de ces problèmes. Pour le problème (a), nous utilisons une spécification *locale* (phrase par phrase) de $\mathcal{Y}$ incluant : les lemmes de la traduction, les gloses lexicales non-présentes dans la traduction, des associations fréquentes listées dans un dictionnaire. Pour ce qui concerne le problème (b), nous exploitons des alignements déterministes obtenus avec SimAlign (section 2.2) pour superviser l'apprentissage (cf. figure 2). Enfin, pour (c), nous augmentons à l'inférence l'espace de recherche avec d'autres mots-candidats, en nous basant sur un dictionnaire issu des données d'entraînement. Notre implémentation s'inspire de Lost (Lavergne *et al.*, 2011, 2013), un modèle probabiliste conçu pour la traduction automatique, qui permet de spécifier localement l'espace de recherche associé à un modèle globalement normalisé (section 2.3).

2.2 Supervision des alignements de gloses lexicales avec SimAlign

L'étape d'apprentissage du modèle CRF étendu demande de définir les étiquetages possibles pour chaque phrase, et au sein de cet ensemble, ceux qui sont jugés corrects. Selon [H2], nous supposons

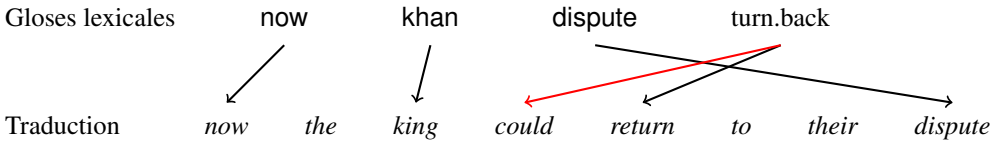
que les étiquettes lexicales constituent un sous-ensemble des mots [3] apparaissant dans la traduction. Les étiquettes correctes (l'association entre morphèmes sources et mots de la traduction) ne sont pas observées : dans ce travail, nous les calculons de manière automatique en exploitant les gloses lexicales disponibles (voir exemple en figure 3) par alignement mot-à-mot entre glose et traduction.

SimAlign L'alignement automatique de mots est réalisé par SimAlign (Jalili Sabet *et al.*, 2020), un modèle d'alignement neuronal basé sur la similarité entre plongements lexicaux de mots. SimAlign implante plusieurs méthodes pour obtenir un alignement à partir d'une matrice de similarité :

— `Argmax` aligne deux mots s'ils sont mutuellement plus proches voisins ;
— `Match` considère l'alignement comme un problème de couplage (*matching*) maximal dans le graphe biparti des mots sources et cibles, en utilisant les similarités pour pondérer les arêtes.

Ces deux méthodes produisent des alignements symétriques, dans lesquels chaque mot de la glose est associé au plus à un mot de la traduction (et réciproquement). Comme nous travaillons sur des gloses et des traductions en anglais, nous utilisons le modèle BERT anglais (BERT-base) (Devlin *et al.*, 2019) pour calculer les similarités. Nos expériences préliminaires montrent que les plongements lexicaux à la sortie de la couche 0 aboutissent au meilleur alignement glose-traduction, probablement dû à l'ordre très différent des gloses lexicales par rapport à l'anglais et l'absence de mots outils dans les gloses.

Comparaison des deux méthodes `Match` identifie par construction un alignement pour chaque glose lexicale [4] et conduit donc à un score de rappel élevé. En comparaison, `Argmax` propose moins de liens d'alignement, mais avec une meilleure précision. Des statistiques sur les alignements ainsi obtenus sont données en section 3.1.



source	howži	xan	yizi–z	daʕba–ƛ'or	uti–ɬ–n
référence	now	**khan**	DEM2.IPL.OBL–GEN2	dispute–SUPER.LAT	**turn.back**–POT–PST.UNW
`Match`	now	**king**	DEM2.IPL.OBL–GEN2	dispute–SUPER.LAT	**could.return**–POT–PST.UNW
`Argmax`	now	**king**	DEM2.IPL.OBL–GEN2	dispute–SUPER.LAT	**return**–POT–PST.UNW

FIGURE 3 – Exemple d'alignements calculés par SimAlign (haut ; heuristique `Argmax` en noir, rouge et noir pour `Match`) et d'étiquettes obtenues en les utilisant (bas). Les différences sont en **gras**.

La figure 3 présente un exemple de phrase étiquetée en utilisant les alignements obtenus par ces deux méthodes. La différence principale réside dans les deux gloses lexicales khan et turn.back qui ne sont pas présentes dans la traduction. Pour la première, SimAlign trouve correctement l'association entre khan et king, ce qui permet d'obtenir une étiquette de sens similaire depuis la traduction. Dans le second cas, `Match` identifie deux liens pour turn.back [5], alors que `Argmax` n'en identifie qu'un.

3. L'alignement est effectué avec la traduction telle quelle, mais les étiquettes associées correspondent aux lemmes des mots identifiés par alignement.
4. Sauf quand la traduction comporte moins d'unités que de gloses lexicales (soit 31 phrases dans nos expériences).
5. Lors de l'alignement, le « . » est enlevé, d'où les deux liens d'alignement. Ce choix nous a permis d'obtenir de meilleurs résultats d'alignements dans des expériences préliminaires.

Cas des gloses non-alignées Dans les cas où SimAlign n'identifie pas de liens d'alignement pour une glose, nous attribuons une étiquette dédiée (« unk »). Cette dernière pouvant concerner une proportion non-négligeable dans le corpus, en particulier dans le cas de `Argmax` (voir section 3.1), nous complétons ces alignements en exploitant un dictionnaire. Ce dernier associe à tout morphème *source*, l'étiquette lexicale qui lui est la plus souvent associée dans la base d'entraînement. Ce « lexique » est ensuite utilisé pour éventuellement attribuer une étiquette lorsque l'alignement fait défaut. Nous assignons ainsi aux morphèmes soit les lemmes des mots de la traduction, soit, en l'absence d'alignement, des étiquettes supplémentaires obtenues via le dictionnaire (cf. figure 2).

2.3 Mise en œuvre de Lost pour la prédiction de gloses

Notre implémentation du modèle probabiliste pour la prédiction de gloses repose sur le système Lost (Lavergne *et al.*, 2011, 2013) conçu originellement dans un cadre de traduction statistique.

Lost est un modèle de traduction à base de segments qui étend les CRF de manière à prendre en charge de très grands ensembles d'étiquettes et de caractéristiques, afin de pouvoir associer aux segments sources des étiquettes correspondant à leurs traductions en langue cible. Dans cette implémentation, il est possible de circonscrire l'ensemble des étiquetages possibles pour chaque phrase, sur la base d'un ensemble réduit d'étiquettes pour chaque mot à annoter. Limiter l'espace de recherche rend l'apprentissage computationnellement faisable en simplifiant le calcul de $Z_\theta(\boldsymbol{x})$ dans l'équation (1).

Espace de recherche Selon [H2], l'espace de recherche est restreint à l'ensemble des étiquettes grammaticales observées à l'entraînement, complété par des lemmes des mots dans la traduction de la phrase traitée (cf. figure 2). À l'apprentissage, nous ajoutons à cet ensemble les gloses des morphèmes non-alignés pour que toute phrase de référence soit atteignable ; à l'inférence, les gloses de référence sont inconnues, et nous ajoutons pour chaque morphème connu l'étiquette lexicale la plus fréquemment associée dans l'ensemble d'apprentissage, conduisant à mettre en œuvre un croisement des hypothèses [H1] & [H2].

Étiquettes simples Notre première configuration utilise la partie supérieure des caractéristiques présentées dans le tableau 1. Cette configuration est illustrée dans la figure 4 : en entrée, Lost reçoit pour chaque morphème source m, sa position t au sein du mot et sa longueur l ; en sortie le modèle prédit uniquement la glose g, supervisée par les liens d'alignement (section 2.2).

Étiquettes structurées La configuration avec étiquettes structurées enrichit la représentation de la sortie de deux informations : d'une part, une étiquette binaire b (GRAM ou LEX) indiquant la nature de la glose, d'autre part, l'étiquette en partie du discours (PoS) p associée au mot aligné dans la phrase traduite. Ces deux informations, qui dérivent de manière déterministe de l'étiquette de base, permettent de construire des caractéristiques supplémentaires (cf. partie inférieure du tableau 1) dont l'estimation est plus robuste [6]. Elles permettent également de généraliser l'étiquetage à des morphèmes inconnus.

6. Pour les étiquettes lexicales issues du dictionnaire, nous utilisons le PoS qui lui est le plus fréquemment associé dans le corpus d'apprentissage.

Caractéristique	Test	Exemple (cf. figure 4, $i = 4$))
uni-gloss	$\mathbb{1}(g_i = g)$	PST.UNW
bi-gloss	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(g_{i-1} = g')$	(say, PST.UNW)
uni-gloss-morph	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(m_i = m)$	n
uni-gloss-position	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(t_i = t)$	1
uni-gloss-length	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(l_i = l)$	1
bi-gloss-morph	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(g_{i-1} = g') \wedge \mathbb{1}(m_i = m)$	(say, PST.UNW, n)
uni/bi-bin	$\mathbb{1}(b_i = b)$ $(\wedge \mathbb{1}(b_{i-1} = b'))$	GRAM ((LEX, GRAM))
uni/bi-pos	$\mathbb{1}(p_i = p)$ $(\wedge \mathbb{1}(p_{i-1} = p'))$	GRAM ((VERB, GRAM))
uni-bin-morph/position/length	$\mathbb{1}(b_i = b) \wedge \mathbb{1}(m_i = m)/\mathbb{1}(t_i = t)/\mathbb{1}(l_i = l)$	(GRAM, n) / (GRAM, 1) / (GRAM, 1)
bi-bin-gloss	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(b_{i-1} = b')$	(LEX, PST.UNW)
bi-gloss-bin	$\mathbb{1}(b_i = b) \wedge \mathbb{1}(g_{i-1} = g')$	(say, GRAM)
uni-pos-morph	$\mathbb{1}(p_i = p) \wedge \mathbb{1}(g_i = g)$	(GRAM, n)
bi-pos-gloss	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(p_{i-1} = p')$	(VERB, PST.UNW)
bi-gloss-pos	$\mathbb{1}(p_i = p) \wedge \mathbb{1}(g_{i-1} = g')$	(say, GRAM)

TABLE 1 – Caractéristiques unigrammes et bigrammes pour le modèle avec étiquettes simples (haut) et structurées (haut + bas).

i	Entrées			Sorties			Référence
	morphème source m	position (mot) t	longueur l	glose approximée g	GRAM ou LEX b	étiquette PoS p	glose de référence
0	q'orol	0	6	widow	LEX	NOUN	widowed
1	ɣˤanabi	0	7	the	LEX	DET	woman
2	a	1	1	ERG	GRAM	GRAM	ERG
3	eʎi	0	3	say	LEX	VERB	say
4	n	1	1	PST.UNW	GRAM	GRAM	PST.UNW

FIGURE 4 – Exemple de configuration pour « q'orol ɣˤanabi–a eʎi–n » (*the widow said*). L'espace de recherche est défini localement par $\mathcal{Y} = \{\text{étiquettes grammaticales}\} \bigcup \{\text{the, widow, say, unk}\}$.

2.4 Conditions expérimentales

Langue Nous étudions principalement le tsez, une langue nakho-daghestanienne parlée dans la république du Daghestan en Russie. Elle est actuellement en cours de documentation (Comrie & Polinsky, à paraître).

Le corpus sur lequel nous travaillons est constitué de récits du folklore, entièrement glosés et traduits en anglais et en russe (Abdulaev & Abdulaev, 2010). Il est constitué de 2 000 phrases avec 40 229 occurrences de morphèmes et 1 603 types de morphèmes (ce qui correspond à 20 153 occurrences de mots). Il existe 157 gloses grammaticales représentant 54 % des étiquettes du corpus et 1 363 gloses lexicales. Dans la mesure où les étiquettes lexicales correspondent à des lemmes en langue anglaise, nous utilisons les traductions en anglais pour calculer les gloses. Notons toutefois que le modèle d'alignement est lui multilingue et pourrait s'appuyer sur les traductions en russe. Ce corpus a, par ailleurs, déjà été étudié dans (Zhao *et al.*, 2020) pour la même tâche [7]. Le tsez fait également partie des langues considérées par le défi organisé en 2023 et visant à évaluer le calcul automatique de gloses interlinéaires [8].

7. Les auteurs ayant un autre découpage du texte (1 782 phrases au total), leurs résultats ne se comparent pas directement aux nôtres. À titre indicatif, ces auteurs obtiennent une correction de 87 avec leurs modèles neuronaux et le modèle à base de CRF de (McMillan-Major, 2020) utilisé comme *baseline*, obtient un score de 84.

8. https://github.com/sigmorphon/2023GlossingST.

Nous divisons le corpus en trois sous-parties cohérentes : 200 phrases pour les jeux de développement et de test, et un jeu d'entraînement de taille variable (200, 500, 1 000 et 1 600), ce qui nous permet d'étudier l'évolution des performances du modèle en fonction des données disponibles. Nous conservons l'ordre des phrases dans le corpus afin de préserver la cohérence des récits.

Métrique Nous donnerons principalement le taux de correction des étiquettes, à savoir la proportion d'étiquettes correctement prédites. Dans nos analyses, nous nous intéressons aussi au rappel différencié selon la nature du morphème (grammaticale ou lexicale).

Paramétrage Nous avons utilisé les paramètres par défaut de Lost. Si Lost permet une régularisation *elastic net*, nous n'utilisons que la régularisation l_1 (paramétrage par défaut avec un poids de $0{,}5$), qui, d'après les expériences préliminaires, semble suffisante. Nous stoppons l'apprentissage après 15 itérations complètes. Pour la lemmatisation et la génération des étiquettes de PoS des traductions en anglais, nous utilisons spaCy [9].

Baseline Le modèle de référence (`maj`) affecte l'étiquette majoritaire observée dans la base d'apprentissage. Comme il n'utilise que les associations entre morphème source et glose déjà vus, il reproduit en quelque sorte le système de « lexique » présent dans certains outils d'annotation tels que ELAN-CorpA ou FieldWorks Language Explorer (FLEx) (Rogers, 2010).

3 Résultats expérimentaux

Dans cette section, nous cherchons tout d'abord à vérifier la validité de l'hypothèse [H2], à savoir la présence des gloses lexicales dans la traduction (section 3.1). Ensuite, nous comparons les performances des configurations, de manière générale et séparée sur les deux catégories de gloses, puis évaluons l'impact de l'alignement et du dictionnaire sur les prédictions (section 3.2).

3.1 Statistiques sur les étiquettes obtenues par alignement

Nous étudions tout d'abord les étiquettes obtenues par alignement. Quelle proportion de gloses lexicales parvient-on à aligner automatiquement ? De plus, nous supposons que les gloses lexicales peuvent être retrouvées dans la traduction : dans quelle mesure est-ce vérifié dans nos projections d'alignements ?

Couverture des gloses obtenues par alignement Le tableau 2 présente le nombre (d'occurrences) et la proportion [10] d'étiquettes lexicales non-alignées avec les deux méthodes de SimAlign. La méthode `Match`, par construction, aligne (presque) toutes les gloses lexicales ; l'apport d'un lexique y est donc limité. En revanche, pour la méthode `Argmax`, qui laisse non-alignées près de 20 % de gloses lexicales, l'utilisation du dictionnaire est essentielle et permet d'aligner plus de 95% des morphèmes dès que l'on dispose de 500 phrases d'entraînement.

9. https://spacy.io/, *pipeline* `en_core_web_sm`.
10. Il y a 18,635 gloses lexicales au total.

Taille du corpus	base /	+ dictionnaire 200	500	1 000	1 600
`Argmax`	3 615 (19,4 %)	1 223 (6,6 %)	858 (4,6 %)	733 (3,9 %)	627 (3,4 %)
`Match`	35 (0,2 %)	18 (0,1 %)	16 (0,1 %)	9 (0,0 %)	9 (0,0 %)

TABLE 2 – Occurrences (et proportion) de gloses lexicales non-alignées par SimAlign et éventuellement complétées par un dictionnaire créé à partir d'un corpus d'apprentissage.

Comparaison avec les gloses de référence Le tableau 3 présente la proportion d'étiquettes obtenues par alignement qui correspondent exactement aux gloses de référence : 80 % des gloses de référence sont prédictibles directement en se basant sur des projections d'alignement depuis la traduction. Comme nous mesurons des correspondances exactes, il est probable que ce taux serait plus élevé si l'on tenait compte des proximités sémantiques (comme khan et king dans l'exemple de la figure 3). Ce premier résultat conforte l'hypothèse [H2].

Dans le cas de base (sans dictionnaire), la méthode `Match` obtient bien une valeur bien plus élevée qu'`Argmax`. La différence entre les méthodes reste toutefois faible au regard du nombre d'alignements supplémentaires identifiés par `Match` (cf. tableau 2), qui correspondent souvent à des erreurs (the / woman figure 4) plutôt qu'à des mots sémantiquement proches.

En utilisant le dictionnaire créé avec les données d'entraînement, toutefois, la tendance s'inverse : si la qualité des étiquettes stagne avec `Match`, elle s'améliore significativement avec `Argmax`. Bien que certains morphèmes restent non-alignés, l'emploi d'un dictionnaire permet néanmoins de se rapprocher des données de référence, tout en restant compatible avec les conditions du test. Dans notre approche, il reste toujours possible que certains morphèmes reçoivent l'étiquette conventionnelle « unk » à l'apprentissage ou au test, leur désambiguïsation restant alors à la charge d'un annotateur.

Taille du corpus	base /	+ dictionnaire 200	500	1 000	1 600
`Argmax`	80,7	84,3	85,1	85,0	85,2
`Match`	81,5	81,5	81,5	81,5	81,6

TABLE 3 – Correspondances exactes entre les gloses de référence et celles obtenues par alignement.

3.2 Résultats de l'étiquetage

Le tableau 4 présente le score de correction des différentes configurations présentées en section 2.3 et le tableau 5 détaille le rappel pour les deux catégories de gloses. Nous remarquons tout d'abord que la *baseline* `maj` est dans nos conditions expérimentales, un modèle d'étiquetage compétitif, avec les meilleurs résultats au niveau des gloses lexicales, pour toute taille de corpus d'entraînement.

L'utilisation du modèle probabiliste permet toutefois de mieux prédire les gloses grammaticales, qui sont plus ambiguës, et s'avère globalement meilleur que la *baseline* à partir de 1 000 phrases environ. Concernant Lost, les configurations avec dictionnaire s'avèrent toujours meilleures que leur équivalent sans dictionnaire. Ceci illustre le bénéfice de l'augmentation de l'espace de recherche par des lemmes probables (d'après le morphème source) mais absents de la traduction, comme en témoigne également la hausse du rappel pour les gloses lexicales (cf. tableau 5). L'utilisation d'étiquettes structurées

Configuration		maj	base simple	base struct.	dict simple	dict struct.
Argmax	200	**64,1**	54,0	53,9	57,7	59,2
	500	**72,5**	62,4	63,5	69,6	71,3
	1000	74,9	67,3	68,3	73,6	**75,3**
	1600	77,1	69,6	71,1	77,1	**77,8**
Match	200	**64,1**	55,5	56,3	59,2	59,7
	500	**72,5**	64,1	65,8	69,3	71,0
	1000	74,9	68,6	69,5	74,1	**76,0**
	1600	77,1	71,0	72,0	77,4	**78,1**

TABLE 4 – Évolution de la correction selon la taille des données d'entraînement. Pour comparer avec chaque méthode d'alignement, les résultats de maj (qui lui n'en dépend pas) sont répétés.

Configuration		P^0_{lex}	maj gram	maj lex	base simple gram	base simple lex	base structuré gram	base structuré lex	dict simple gram	dict simple lex	dict structuré gram	dict structuré lex
Argmax	200	32,3 %	71,2	**56,1**	**81,6**	23,4	80,0	25,1	78,5	34,7	79,8	36,4
	500	16,9 %	72,4	**72,7**	85,8	36,5	**86,5**	38,0	83,9	53,8	86,3	54,8
	1000	9,9 %	72,8	**77,3**	89,2	42,9	89,4	45,0	88,7	56,9	**90,0**	59,1
	1600	4,5 %	73,0	**81,6**	90,5	46,5	**91,0**	49,0	90,1	62,7	90,8	63,5
Match	200	32,3 %	71,2	**56,1**	79,3	29,1	79,2	30,9	**80,5**	35,6	80,1	37,2
	500	16,9 %	72,4	**72,7**	84,9	41,1	86,4	43,1	85,0	51,9	**87,0**	53,3
	1000	9,9 %	72,8	**77,3**	88,2	46,9	88,8	48,0	88,2	58,5	**89,2**	61,3
	1600	4,5 %	73,0	**81,6**	89,2	50,8	**90,3**	51,6	88,9	64,7	89,8	65,1

TABLE 5 – Scores de rappel différenciés selon la nature de la glose. P^0_{lex} indique la proportion d'étiquettes *lexicales* présentes à l'*inférence* mais jamais observées à l'*entraînement*.

améliore sensiblement les prédictions, illustrant l'intérêt des caractéristiques supplémentaires faisant intervenir des catégories plus générales (et robustes) comme GRAM, LEX ou les PoS. Enfin les deux méthodes d'alignement de SimAlign sont très proches, avec un petit avantage pour Match. Une explication est que cette méthode génère moins d'étiquettes unk qu'Argmax ; par exemple, pour la configuration utilisant des étiquettes structurées et un dictionnaire, le premier produit environ 3 % d'étiquettes inconnues, contre 7 % pour le second.

Il faut finalement rappeler que les résultats du tableau 4 évaluent les correspondances *exactes* entre les gloses de référence et les prédictions et sous-estiment la qualité des gloses automatiques. Ceci est illustré par la figure 3 où les modèles basés sur Lost proposent king, pourtant comptabilisé comme une erreur car différant de la référence (kahn).

4 Conclusion

Dans cet article, nous avons présenté une nouvelle approche pour aborder la tâche de génération automatique de gloses linéaires. D'une part, après avoir validé l'hypothèse que les gloses lexicales sont le plus souvent présentes dans les traductions, nous avons utilisé des alignements automatiques pour superviser l'apprentissage d'un modèle de glose, en complétant éventuellement avec un dictionnaire. D'autre part, nous avons eu recours à une extension du modèle CRF, Lost, permettant de restreindre

et sélectionner les gloses lexicales possibles pour une phrase donnée. Dans nos conditions expérimentales, nous parvenons alors à surpasser le modèle de base avec 1 000 phrases d'entraînement, l'apprentissage améliorant en particulier les gloses grammaticales.

Ces résultats sous-estiment la qualité des gloses automatiques, du fait de la mesure considérée : pour quantifier cette sous-estimation, nous prévoyons de réaliser un alignement manuel entre source et traduction, afin de disposer de références qui correspondent exactement au problème d'apprentissage traité. Pour le futur, plusieurs pistes sont explorées, en particulier l'utilisation des jeux de caractéristiques plus riches et l'exploration des manières alternatives de construire l'espace de recherche pour l'étiquetage. Il serait par exemple intéressant d'exploiter également des alignements entre mots/morphèmes grammaticaux (en source et en cible) qui pourraient aider à mieux localiser les morphèmes pleins qui leur sont proches. Pour améliorer l'apprentissage, il est aussi envisagé de relâcher les contraintes découlant de l'utilisation d'un alignement (ici, calculé par SimAlign) et d'apprendre le modèle probabiliste avec des alignements latents. Une autre perspective est d'augmenter la base de données utilisées à l'apprentissage en combinant des corpus documentant plusieurs langues : l'objectif sera alors de faire émerger des caractéristiques « multilingues » dans les langues documentées pour améliorer la robustesse de caractéristiques testant des propriétés génériques (par exemple, la longueur ou la position relative des morphèmes grammaticaux vs. non-grammaticaux).

Remerciements

Ce travail est effectué dans le cadre du projet franco-allemand ANR-DFG « La documentation automatique des langues à l'horizon 2025 » (*Computational Language Documentation by 2025*, CLD 2025, ANR-19-CE38-0015-04). Les auteurs remercient chaleureusement Thomas Lavergne pour l'accompagnement dans les expériences avec Lost, ainsi qu'Antonios Anastasopoulos pour la mise à disposition du corpus tsez.

Références

ABDULAEV A. K. & ABDULAEV I. K. (2010). *Cezjas fol'klor : (gíurus mecrek° iorno butirno) = Dido (Tsez) folklore = Didojskij (cezskij) fol'klor*. Leipzig : Lotos.

BALDRIDGE J. & PALMER A. (2009). How well does active learning *actually* work ? Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, p. 296–305, Singapore : Association for Computational Linguistics.

BARRIGA MARTÍNEZ D., MIJANGOS V. & GUTIERREZ-VASQUES X. (2021). Automatic interlinear glossing for Otomi language. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, p. 34–43, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.americasnlp-1.5.

BICKEL B., COMRIE B. & HASPELMATH M. (2008). The Leipzig Glossing Rules : Conventions for interlinear morpheme-by-morpheme glosses. Leipzig : Max Planck Institute for Evolutionary Anthropology, Department of Linguistics. https://www.eva.mpg.de/lingua/resources/glossing-rules.php.

COMRIE B. & POLINSKY M. (à paraître). Tsez. In Yuri Koryakov, Yury Lander and Timur Maisak (eds.) The Caucasian Languages. An International Handbook. Mouton. HSK series.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

JALILI SABET M., DUFTER P., YVON F. & SCHÜTZE H. (2020). SimAlign : High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 1627–1643, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.147.

LAFFERTY J. D., MCCALLUM A. & PEREIRA F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, p. 282–289, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

LAVERGNE T., ALLAUZEN A., CREGO J. M. & YVON F. (2011). From n-gram-based to CRF-based translation models. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, p. 542–553, Edinburgh, Scotland : Association for Computational Linguistics.

LAVERGNE T., ALLAUZEN A. & YVON F. (2013). Un cadre d'apprentissage intégralement discriminant pour la traduction statistique. In *Actes de la 20ème Conférence sur le Traitement Automatique des Langues Naturelles*, p. 450–463, Les Sables d'Olonne, France : ATALA. HAL : hal-01908381.

LAVERGNE T. & YVON F. (2017). Learning the structure of variable-order CRFs : a finite-state perspective. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2017, p. 433–439 : Association for Computational Linguistics. DOI : 10.18653/v1/D17-1044.

MCMILLAN-MAJOR A. (2020). Automating gloss generation in interlinear glossed text. In *Proceedings of the Society for Computation in Linguistics 2020*, p. 355–366, New York, New York : Association for Computational Linguistics.

MOELLER S. & HULDEN M. (2018). Automatic glossing in a low-resource setting for language documentation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, p. 84–93, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

MUELLER T., SCHMID H. & SCHÜTZE H. (2013). Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 322–332, Seattle, Washington, USA : Association for Computational Linguistics.

OKABE S. & YVON F. (2022). Vers la génération automatique de gloses pour la documentation automatique des langues. In L. BECERRA, B. FAVRE, C. GARDENT & Y. PARMENTIER, Éds., *Journées Jointes des Groupements de Recherche Linguistique Informatique, Formelle et de Terrain (LIFT) et Traitement Automatique des Langues (TAL)*, p. 198–203, Marseille, France : CNRS. HAL : hal-03846843.

ROGERS C. (2010). Review of Fieldworks Language Explorer (FLEx) 3.0. In *Language Documentation & Conservation 4*, p. 78–84.

SAMARDŽIĆ T., SCHIKOWSKI R. & STOLL S. (2015). Automatic interlinear glossing as two-level sequence classification. In *Proceedings of the 9th SIGHUM Workshop on Language Technology*

for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), p. 68–72, Beijing, China : Association for Computational Linguistics. DOI : 10.18653/v1/W15-3710.

SEIFART F., EVANS N., HAMMARSTRÖM H. & LEVINSON S. (2018). Language documentation twenty-five years on. *Language*, **94**(4), e324–e345. DOI : 10.1353/lan.2018.0070.

TELLIER I. & TOMMASI M. (2011). Champs Markoviens Conditionnels pour l'extraction d'information. In E. GAUSSIER & F. YVON, Éds., *Modèles probabilistes pour l'accès à l'information textuelle*, p. 223–267. Hermès. HAL : inria-00514525.

ZHAO X., OZAKI S., ANASTASOPOULOS A., NEUBIG G. & LEVIN L. (2020). Automatic interlinear glossing for under-resourced languages leveraging translations. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 5397–5408, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.471.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Intégration de connaissances structurées par synthèse de texte spécialisé

Guilhem Piat[1,2] Ellington Kirby[1] Julien Tourille[2]
Nasredine Semmar[2] Alexandre Allauzen[1] Hassane Essafi[2]
(1) Université Paris Dauphine, F-75775, Paris Cedex 16, France
(2) Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{prenom}.{nom}@dauphine.psl.eu, {prenom}.{nom}@cea.fr

RÉSUMÉ

Les modèles de langue de type Transformer peinent à incorporer les modifications ayant pour but d'intégrer des formats de données structurés non-textuels tels que les graphes de connaissances. Les exemples où cette intégration est faite avec succès requièrent généralement que le problème de désambiguïsation d'entités nommées soit résolu en amont, ou bien l'ajout d'une quantité importante de texte d'entraînement, généralement annotée. Ces contraintes rendent l'exploitation de connaissances structurées comme source de données difficile et parfois même contre-productive. Nous cherchons à adapter un modèle de langage au domaine biomédical en l'entraînant sur du texte de synthèse issu d'un graphe de connaissances, de manière à exploiter ces informations dans le cadre d'une modalité maîtrisée par le modèle de langage.

ABSTRACT

Knowledge Integration by In-domain Text Generation

Transformer-based language models have trouble integrating modifications whose purpose is to incorporate knowledge from structured, non-textual data such as knowledge graphs. Instances where this integration is successful generally require the problem of Entity Linking to be solved upstream, or the addition of a significant amount of (generally annotated) text to the training set. These constraints often make leveraging structured data difficult and/or counterproductive. We seek to adapt a language model to the biomedical domain through training on synthetic text derived from a knowledge graph, such that the information therein can be effectively leveraged in a format which the model can handle natively.

MOTS-CLÉS : Intégration de connaissances ; Génération de texte ; Adaptation au domaine ; Modèle de langage biomédical.

KEYWORDS: Knowledge integration ; Text generation ; Domain adaptation ; Biomedical language model.

1 Introduction et travaux connexes

En raison du vocabulaire spécialisé et de la spécificité des concepts traités dans certains domaines de spécialité, les performances des modèles de langage pré-entraînés sur du texte général, tels que BERT (Devlin *et al.*, 2019), ont tendance à souffrir dans ces domaines. Diverses approches de spécialisation existent, la plus notable et évidente d'entre elles étant la spécialisation par pré-

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 275–284.

entraînement sur du texte issu du domaine ciblé. La performance de cette approche a été étudiée par Gururangan *et al.* (2020) et El Boukkouri (2021), et de nombreux modèles ont été couronnés de succès en suivant cette méthode, comme par exemple LegalBERT (Chalkidis *et al.*, 2020) dans le domaine légal et PatentBERT (Lee & Hsiang, 2020) dans le domaine des brevets. Dans le domaine biomédical, sur lequel nous décidons de nous focaliser (bien que les concepts dont nous traitons soient *a priori* applicables à tout domaine de spécialité), une variété de modèles appliquent la même méthode générale, comme BioBERT (Lee *et al.*, 2019), BlueBERT (Peng *et al.*, 2019) ou PubMedBERT (Gu *et al.*, 2021). Cette approche n'est cependant pas sans défauts. En particulier, elle est susceptible à l'oubli catastrophique (Xu *et al.*, 2020), et requiert une quantité importante de texte issu du domaine, ce qui n'est pas une option disponible dans tous les domaines de spécialité. De surcroît, comme observé par Zipf (1935), la distribution des mots dans langage naturel est fortement biaisée, ce qui a pour conséquence qu'accroître le nombre de concepts couverts par le corpus nécessite une augmentation exponentielle de la quantité de texte.

Une classe concurrente d'approches consiste à intégrer des informations issues d'une base de connaissance spécialisée. Diverses stratégies ont été mises en œuvre, comme l'utilisation du mécanisme d'attention des modèles *Transformer* pour combiner les informations issues des mots avec celles issues des entités (par ERNIE (Zhang *et al.*, 2019), KnowBert (Peters *et al.*, 2019), K-Adapter (Wang *et al.*, 2021a) et DRAGON (Yasunaga *et al.*, 2022) en particulier), l'alignement de plongements de mots et de plongements d'entités (KEPLER (Wang *et al.*, 2021b), CODER (Yuan *et al.*, 2022)), ou encore le changement de la fonction de coût en entraînement de manière à modéliser explicitement la synonymie des termes spécialisés (UmlsBERT (Michalopoulos *et al.*, 2021)).

Bien que ces approches puissent donner des résultats remarquables, leur champ d'applicabilité est limité par le fait qu'elles ne permettent pas (ou peu) de réduction de la quantité de texte nécessaire à la spécialisation, et souvent même requièrent du texte aux entités annotées ; or ces ressources ne sont pas disponibles pour tous les domaines. De plus, la majorité des approches requièrent, en entraînement comme, généralement, en déploiement, que les entités de la base de connaissance soient identifiées dans le texte pour fonctionner. Or, ce problème n'est pas résolu pour tous les domaines, et les meilleurs outils actuels sont gourmands en temps de calcul, limitant l'échelle à laquelle ils peuvent être déployées. A l'heure d'écriture, l'état de l'art en désambiguïsation d'entités nommées biomédicales (Bhowmik *et al.*, 2021) atteint une F-mesure de $0,564$ sur le corpus MedMentions (Mohan & Li, 2019).

Nous cherchons donc une méthode d'intégration de connaissances telle qu'aucun texte du domaine cible et aucune désambiguïsation — en entraînement comme en inférence — ne soient nécessaires. La combinaison de modalités différentes débouchant invariablement sur un besoin de données d'entraînement supplémentaires, nous cherchons à exploiter notre base de connaissance sous forme textuelle. Nous proposons une procédure permettant de générer un corpus contenant une phrase par paire de concepts liés par une relation dans une base de connaissance. Il est attendu qu'un corpus produit ainsi soit plus dense en informations factuelles liées au domaine que du texte naturel, et couvrira une plus grande variété de concepts pour le même volume de texte, la fréquence d'apparition des concepts étant dictée par la topologie du graphe de connaissances et non l'usage du langage naturel.

2 Méthode

2.1 Graphe de connaissances

Nous nous plaçons dans le cas où les nœuds de notre graphe de connaissances représentent des concepts et les arêtes des relations. Nous pouvons donc extraire des triplets (c_i, r, c_j) tels qu'il existe un lien sémantique entre les concepts c_i et c_j qui peut être formulé en langage naturel. Spécifiquement, nous cherchons à associer des groupes nominaux N_i à c_i et N_j à c_j, ainsi qu'un syntagme verbal V_r à r de sorte que la phrase $N_i V_r N_j$ soit une phrase grammaticalement correcte qui capture le sens de la relation dans le graphe. La base de connaissance peut ensuite être représentée par un ensemble de phrases factuelles, exploitable comme corpus d'apprentissage par un modèle de langue.

Nous utilisons le graphe de connaissances UMLS, dans sa version 2022AB, avec toutes et seulement les ressources anglophones. Il existe dans notre version de la base de connaissance 1008 types de relations et environ $4,6$ millions d'entités distinctes pour un total d'environ $39,7$ millions de triplets. Chaque concept dispose d'un identifiant unique c et d'un ou plusieurs « *noms* » $N_c^1, \ldots, N_c^{k_c}$, groupes nominaux qui correspondent à des formulations récurrentes du concept en langage naturel, et dont un est considéré *préféré*. Chaque type de relation est représenté par une courte chaîne de caractères r proche du langage naturel, décrivant grossièrement la relation. Le graphe UMLS est représenté sous forme de base de données relationnelle, avec une table contenant les triplets d'intérêt, représentés de la manière suivante : c_i, r, c_j.

2.2 Génération de texte

Bien qu'il y ait un bénéfice potentiel évident à inclure, le cas échéant, plusieurs formulations pour chaque entité, plusieurs facteurs nous ont menés à conserver seulement le *nom préféré* de chaque concept :

- le niveau de redondance entre les noms est élevé, contenant principalement des doublons, des variations de casse et des variations d'accord au pluriel.
  ex. : Le concept pour l'ADN a 88 noms, dont 20 occurrences de *DNA*, 15 variations de casse et d'accord en nombre sur *Deoxyribonucleic Acid*, 4 variations de casse et d'accord sur *DNA molecule*, et 10 combinaisons des termes susmentionnés avec des ordres et ponctuations différents.
- Chaque triplet étant constitué de deux concepts, le temps de calcul augmente quadratiquement avec le nombre total de noms lors de la résolution des triplets sous la forme (c_j, r, c_i) à la forme $(N_{c_i}^{1,\ldots,k_{c_i}}, r, N_{c_j}^{1,\ldots,k_{c_j}})$.
- Inclure les noms non-préférés réduit l'homogénéité grammaticale des groupes nominaux (par ex., la grande majorité des noms préférés sont au singulier), complexifiant la tâche de générer des phrases grammaticalement correctes.
- Multiplier les occurrences des concepts associés à de nombreux noms induit un biais dans l'apprentissage qui n'est pas forcément désirable.

Étant donné le nombre important de triplets, la résolution des noms n'est pas traitable sur l'ensemble de la base de connaissances. De plus, tous les types de relation ne sont pas utiles ou documentés. Nous effectuons donc une sélection sur les types de relations. Nous commençons par éliminer les 925 relations les plus rares, qui constituent environ 15% des triplets. Ensuite, nous éliminons une relation de chaque paire de relations symétriques, c'est-à-dire que lorsqu'il existe deux relations r_1

et r_2 telles que (c_i, r_1, c_j) représente la même information que (c_j, r_2, c_i), nous ne conservons que la relation r_1. Environ 95% des relations font partie d'une paire symétrique, ce qui nous laisse 43 relations. Nous avons ensuite sélectionné les 28 relations qui avaient un sens apparent ou documenté et qui n'impliquaient pas principalement des concepts aux noms complexes comme des molécules ou protéines.

Une fois nos triplets (N_{c_i}, r, N_{c_j}) extraits de la base de connaissance, nous avons formulé une phrase-type par relation dans laquelle nous avons automatiquement inséré les noms d'entités correspondant aux triplets. Un échantillon des séquences ainsi générées depuis la base de connaissance UMLS se trouve en annexe A. Le corpus résultant est constitué d'environ 6 millions de phrases simples, ou 100 millions de mots. Nous appelons *CSGU* ce Corpus Synthétique Généré depuis UMLS.

2.3 Modèles de langue

Nous entraînons le modèle BERT (Devlin *et al.*, 2019) pré-entraîné, spécifiquement dans sa version $\text{BERT}_{\text{BASE}}$, sur trois corpus différents [1]. Le premier est CSGU, et nous appelons le modèle résultant $\text{BERT}_{\text{CSGU}}$. De manière à évaluer l'efficacité de notre corpus synthétique vis-à-vis d'un corpus naturel, nous avons assemblé un second corpus de texte biomédical que nous appelons PMC, de volume similaire (environ 94 millions de mots) constitué d'une collection d'articles scientifiques en accès libre recueillis depuis PubMed Central. Nous appelons le modèle entraîné sur ce corpus BERT_{PMC}. De manière à étudier l'effet du texte de synthèse comme option d'augmentation de données, nous entraînons un modèle sur un corpus hybride constitué de la concaténation du corpus de synthèse et du corpus naturel. Cependant, l'ajout de données d'entraînement se faisant rarement au détriment du processus d'apprentissage, nous entraînons également un modèle sur la moitié du corpus hybride, de manière à ce que la quantité de données d'entraînement soit comparable aux autres modèles. Nous appelons ces modèles $\text{BERT}^{100\%}_{\text{Hybr}}$ et $\text{BERT}^{50\%}_{\text{Hybr}}$ respectivement. Les informations sur les hyperparamètres des modèles sont disponibles en annexe B.

Nous effectuons une batterie de tests sur chacun des modèles. Nous détaillons ces tests dans la section 3 et présentons les résultats sur les tâches biomédicales et générales dans les Tables 1 et 2 respectivement [2].

3 Résultats expérimentaux

NB : Les tâches marquées par un obèle (†) sont altérées vis-à-vis du standard et les résultats ne sont pas nécessairement comparables à ceux de la littérature.

3.1 Complétion de phrases biomédicales à trous (MLM)

Le but de l'approche par intégration de connaissances étant de permettre au modèle de mieux modéliser les co-occurrences de concepts biomédicaux, nous cherchons à évaluer la capacité de

1. Le code pour l'entraînement des modèles et les informations nécessaires à la création des corpus est disponible sur notre dépôt GitHub.

2. En supplément des résultats présentés ici, nous avons abordé la tâche de reconnaissance d'entités nommées i2b2 2010/n2c2 (Uzuner *et al.*, 2011), mais ne rapportons pas les résultats car les différences de performance n'étaient pas statistiquement significatives.

notre modèle à prédire les concepts masqués dans des contextes de phrases biomédicales. Les textes biomédicaux aux concepts annotés étant cependant rares, nous évaluons nos modèles sur la tâche de modélisation de langue par masquage de mots. Nous appliquons cette tâche à un corpus biomédical d'environ $12,6$ millions de mots recueilli de la même manière que le corpus d'entraînement du modèle BERT_{PMC}. Puisque tous les termes masqués ne seront pas des termes biomédicaux, une bonne performance sur cette tâche est indicative à la fois d'une bonne maîtrise du langage général et de la terminologie biomédicale.

Pour évaluer les performances de nos modèles sur cette tâche, nous utilisons un critère apparenté d'un point de vue théorique à la *perplexité*, définie dans les modèles de langue autorégressifs comme l'exponentielle de la moyenne des log-vraisemblances de la séquence, et équivalente à l'exponentielle de l'entropie croisée entre les données et les prédictions :

$$\exp\left(-\frac{1}{N}\sum_{i=1}^{N}\mathbf{y_i}\cdot\log(\hat{\mathbf{y}}_\mathbf{i})\right) \tag{1}$$

où $\mathbf{y_i}$ et $\hat{\mathbf{y}}_\mathbf{i}$ sont respectivement le label (en encodage *one-hot*) et la distribution de probabilité prédite pour le jeton i, avec N le nombre total de jetons dans la séquence.

BERT n'étant pas un modèle autorégressif, la perplexité n'est pas strictement définie. D'un point de vue pratique cependant, l'équation (1) peut être calculée, et nous donne une mesure de performance cohérente. Par souci de simplicité, nous désignerons donc cette mesure « perplexité », ou « *ppl.* ». Une perplexité *basse* indique une meilleure performance.

Nos résultats rapportés dans la table 1 indiquent que le texte synthétique confère à $\text{BERT}_{\text{CSGU}}$ une capacité prédictive accrue par rapport à $\text{BERT}_{\text{BASE}}$. Il semblerait cependant que le manque d'exposition au langage naturel dans cette phase d'entraînement lui porte préjudice puisque BERT_{PMC} dépasse $\text{BERT}_{\text{CSGU}}$. Les modèles $\text{BERT}_{\text{Hybr}}$, cependant, dépassent le niveau de performance de BERT_{PMC}, indiquant que les défaillances de CSGU sont facilement atténuées par l'ajout de texte naturel dans le corpus.

3.2 Extraction de relations biomédicales (ChemProt†)

La tâche ChemProt (Krallinger *et al.*, 2017) consiste à prédire, étant donné une séquence biomédicale contenant deux entités E_1 et E_2, la nature de leur relation, parmi six options. Nos résultats révèlent que le corpus CSGU contient des informations précieuses, mais les connaissances acquises au cours du processus d'apprentissage sont fragiles et l'inclusion de texte naturel peut être contreproductive dans leur assimilation.

NB : Plusieurs versions du corpus ChemProt existent avec des types de pré-traitement différents qui peuvent influencer les performances des modèles. Tous nos modèles sont entraînés sur la même version du corpus et sont donc comparables entre eux.

3.3 Questions-réponses biomédicales (PubmedQA)

Le corpus PubmedQA (Jin *et al.*, 2019) est associé à plusieurs tâches de questions-réponses. Chaque instance contient une question sous forme de problématique issue d'un article scientifique, plusieurs

séquences dites « de contexte » apportant des éléments de réponse, et une phrase dite de « réponse longue » contenant la conclusion nuancée de l'article. L'objectif est de classer les questions selon la nature affirmative, négative, ou indécise de leur conclusion.

Nous abordons la tâche dans sa version la plus simple, à savoir la version dite « sans raisonnement ». Dans ce cadre, notre processus d'entraînement inclut la question et la réponse longue, mais ignore les séquences de contexte.

Nous constatons d'après nos résultats expérimentaux (Table 1) que $\text{BERT}_{\text{BASE}}$ a des performances bien plus faibles que les modèles spécialisés, ce qui est attendu puisqu'il n'est pas familiarisé avec le vocabulaire biomédical. Les performances supérieures de $\text{BERT}_{\text{CSGU}}$ laissent penser que la maîtrise des relations entre concepts biomédicaux et l'exposition à une grande variété de concepts sont particulièrement utiles pour interpréter des problématiques et conclusions scientifiques, et la dominance de $\text{BERT}^{100\%}_{\text{Hybr}}$ indique que les connaissances apportées par les corpus CSGU et PMC sont complémentaires dans le contexte de cette tâche.

Model	MLM (ppl.)	ChemProt (F_1)	PubmedQA (F_1)
$\text{BERT}_{\text{BASE}}$	16,64	88,91	70,20
BERT_{PMC}	13,66	88,91	74,83
$\text{BERT}_{\text{CSGU}}$	14,67	<u>89,87</u>	75,67
$\text{BERT}^{50\%}_{\text{Hybr}}$	11,38	88,88	72,50
$\text{BERT}^{100\%}_{\text{Hybr}}$	<u>10,37</u>	88,89	<u>78,00</u>

TABLE 1 – Tableau comparatif des résultats des modèles incorporant du texte synthétique dans leur corpus d'entraînement vis-à-vis des modèles entraînés exclusivement sur du langage naturel sur les tâches biomédicales de complétion de phrases à trous (MLM), d'extraction de relations (ChemProt), et de questions-réponses (PubMedQA). Résultats moyens sur 4 expériences. Le résultat souligné est le meilleur pour chaque tâche.

3.4 Tâches non biomédicales (CoLA†, SNLI)

Nous évaluons également nos modèles sur des tâches non biomédicales de manière à évaluer la baisse de performances dans le domaine général encourue par les modèles spécialisés.

La tâche d'acceptabilité linguistique CoLA consiste à classer différentes séquences selon leur qualité grammaticale comme étant « acceptables » ou non. Le critère d'évaluation pour cette tâche est le coefficient de corrélation de Matthews. Les labels de la partition de test n'étant pas publics, et en raison de diverses restrictions de soumission, nous avons re-partitionné le jeu de données. Nous avons utilisé la partition de validation comme partition de test, et de manière à rendre ce partitionnement reproductible, nous avons utilisé les 500 dernières instances de la partition d'entraînement comme partition de validation.

La tâche d'inférence linguistique SNLI est une tâche de classification de paires de séquences selon l'existence d'une relation d'*implication*, de *contradiction*, ou l'*absence* d'une telle relation entre elles.

Nos résultats dans la Table 2 indiquent que l'apprentissage sur le texte de synthèse porte préjudice à la capacité du modèle à évaluer la qualité grammaticale d'une séquence, sans pour autant avoir une incidence majeure sur sa capacité à détecter les relations d'implication. L'affaiblissement en grammaire peut cependant être compensé en associant le texte de synthèse à du langage naturel.

Modèle	CoLA (Corr. M.)	SNLI (F_1)
$\text{BERT}_{\text{BASE}}$	63,17	<u>90,58</u>
BERT_{PMC}	<u>64,11</u>	90,38
$\text{BERT}_{\text{CSGU}}$	61,89	90,44
$\text{BERT}^{50\%}_{\text{Hybr}}$	61,62	90,28
$\text{BERT}^{100\%}_{\text{Hybr}}$	63,86	90,20

TABLE 2 – Tableau comparatif des résultats des modèles incorporant du texte synthétique dans leur corpus d'entraînement vis-à-vis des modèles entraînés exclusivement sur du langage naturel sur les tâches d'acceptabilité linguistique (CoLA), et d'inférence (SNLI). Résultats moyens sur 4 expériences. Le résultat souligné est le meilleur pour chaque tâche.

4 Conclusions et futurs travaux

Nous proposons une procédure d'intégration de connaissances pour l'adaptation des modèles de langage aux domaines de spécialité simple à mettre en œuvre, et qui ne dépend ni de texte issu du domaine, ni d'outils d'annotation d'entités, ni d'une architecture de modèle spécialisée. Nous démontrons que, malgré la qualité dégradée du texte généré par rapport à du texte naturel, il peut être exploité par un modèle de langue pour l'adaptation au domaine avec, pour une quantité de texte fixe, plus de succès que du texte naturel. Enfin, les faiblesses exhibées par les modèles entraînés sur du texte synthétique peuvent être minimisées par l'incorporation, dans le corpus de spécialisation, de texte issu du domaine lorsqu'il est disponible.

Outre l'application de cette méthode à d'autres bases de connaissance comme YAGO (Suchanek *et al.*, 2007) ou WorldKG (Dsouza *et al.*, 2021), nous aimerions à l'avenir intégrer à cette méthode un post-traitement intelligent capable d'identifier les séquences grammaticalement incorrectes, excessivement complexes ou autrement problématiques, et de les supprimer ou les corriger. Par ailleurs, le manque de variété linguistique constitue une faiblesse importante de notre approche. S'il serait sans doute difficile d'automatiser des variations sur la formulation des relations, une amélioration envisageable serait d'intégrer des informations concernant la variété des formulations de concepts, soit en sélectionnant aléatoirement, étant donné un concept c, une formulation parmi $N_c^1, \ldots, N_c^{K_c}$, soit en ajoutant au corpus de synthèse des séquences dédiées à expliciter les synonymes (par exemple : « "N_c^2" is another name for "N_c^1". »)

Enfin, cette méthode d'intégration de connaissances étant applicable à tout modèle de langue, la combiner avec d'autres méthodes comme KnowBert, KEPLER ou DRAGON pourrait être une manière peu coûteuse d'accroître leurs performances, et pourrait établir un nouvel état de l'art dans ce domaine de recherche.

Remerciements

Cette publication a été rendue possible grâce à l'utilisation du supercalculateur FactoryIA, soutenu financièrement par le Conseil Régional d'Ile-De-France.

Références

BHOWMIK R., STRATOS K. & DE MELO G. (2021). Fast and effective biomedical entity linking using a dual encoder. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, p. 28–37.

CHALKIDIS I., FERGADIOTIS M., MALAKASIOTIS P., ALETRAS N. & ANDROUTSOPOULOS I. (2020). LEGAL-BERT : The Muppets straight out of law school. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 2898–2904, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.261.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186.

DSOUZA A., TEMPELMEIER N., YU R., GOTTSCHALK S. & DEMIDOVA E. (2021). Worldkg : A world-scale geographic knowledge graph. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, p. 4475–4484.

EL BOUKKOURI H. (2021). *Domain adaptation of word embeddings through the exploitation of in-domain corpora and knowledge bases*. Thèse de doctorat, Université Paris-Saclay.

GU Y., TINN R., CHENG H., LUCAS M., USUYAMA N., LIU X., NAUMANN T., GAO J. & POON H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, **3**(1), 1–23.

GURURANGAN S., MARASOVIĆ A., SWAYAMDIPTA S., LO K., BELTAGY I., DOWNEY D. & SMITH N. A. (2020). Don't stop pretraining : Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8342–8360 : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.740.

JIN Q., DHINGRA B., LIU Z., COHEN W. & LU X. (2019). Pubmedqa : A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 2567–2577.

KRALLINGER M., RABAL O., AKHONDI S. A., PÉREZ M. P., SANTAMARÍA J., RODRÍGUEZ G. P., TSATSARONIS G., INTXAURRONDO A., LÓPEZ J. A., NANDAL U. *et al.* (2017). Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, p. 141–146.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2019). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, p. btz682. DOI : 10.1093/bioinformatics/btz682.

LEE J.-S. & HSIANG J. (2020). Patent classification by fine-tuning bert language model. *World Patent Information*, **61**, 101965.

MICHALOPOULOS G., WANG Y., KAKA H., CHEN H. & WONG A. (2021). UmlsBERT : Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1744–1753.

MOHAN S. & LI D. (2019). MedMentions : A Large Biomedical Corpus Annotated with UMLS Concepts. *arXiv :1902.09476 [cs]*.

PENG Y., YAN S. & LU Z. (2019). Transfer learning in biomedical natural language processing : An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, p. 58–65.

PETERS M. E., NEUMANN M., LOGAN R. L., SCHWARTZ R., JOSHI V., SINGH S. & SMITH N. A. (2019). Knowledge enhanced contextual word representations. In *EMNLP*.

SUCHANEK F. M., KASNECI G. & WEIKUM G. (2007). Yago : A Core of Semantic Knowledge. In *16th International Conference on the World Wide Web*, p. 697–706.

UZUNER Ö., SOUTH B. R., SHEN S. & DUVALL S. L. (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, **18**(5), 552–556.

WANG R., TANG D., DUAN N., WEI Z., HUANG X.-J., JI J., CAO G., JIANG D. & ZHOU M. (2021a). K-Adapter : Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, p. 1405–1418.

WANG X., GAO T., ZHU Z., ZHANG Z., LIU Z., LI J. & TANG J. (2021b). Kepler : A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, **9**, 176–194.

XU Y., ZHONG X., YEPES A. J. J. & LAU J. H. (2020). Forget me not : Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)*, p. 1–8 : IEEE.

YASUNAGA M., BOSSELUT A., REN H., ZHANG X., MANNING C. D., LIANG P. & LESKOVEC J. (2022). Deep bidirectional language-knowledge graph pretraining. *arXiv preprint arXiv :2210.09338*.

YUAN Z., ZHAO Z., SUN H., LI J., WANG F. & YU S. (2022). CODER : Knowledge-infused cross-lingual medical term embedding for term normalization. *Journal of biomedical informatics*, **126**, 103983.

ZHANG Z., HAN X., LIU Z., JIANG X., SUN M. & LIU Q. (2019). ERNIE : Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 1441–1451.

ZIPF G. K. (1935). *The Psycho-Biology of language*, volume 21. Psychology Press.

Annexes

A Échantillon aléatoire du corpus généré

1. *Skull Fractures is a type of Fracture of bone of head.*
2. *Procedures on breast is a type of Procedure on trunk.*
3. *Tarsum is an ingredient in Coal Tar 20 MG/ML / Salicylic Acid 50 MG/ML Medicated Shampoo [Tarsum].*
4. *The concept of "MK-5108" is part of the CTRP Agent Terminology.*
5. *Multi vessel coronary artery disease is an example of Coronary Arteriosclerosis.*
6. *amitriptyline hydrochloride 25 MG / perphenazine 2 MG [Etrafon] is a brand name for perphenazine 2 MG.*
7. *Trunk of parotid branch of left superficial temporal artery is a type of Trunk of parotid branch of superficial temporal artery.*
8. *Biopsy of lesion of internal nose can be used to identify biopsy of nasal cavity : carcinoid tumor.*
9. *Uterine fibroid embolization is a type of Embolization of artery.*
10. *Meperidine analog-containing product is a type of Piperidine derivative.*

B Hyperparamètres

Nos expériences ont révélé que les hyperparamètres de pré-entraînement optimaux étaient les mêmes d'un modèle à l'autre, ils ne sont donc pas différenciés dans la Table 3. Les hyper-paramètres des tâches d'évaluation ont été optimisés vis-à-vis de $\text{BERT}_{\text{BASE}}$. Il n'y a pas d'entrée pour la tâche MLM car les modèles sont déjà optimisés pour cette tâche en pré-entraînement.

Tâche	Pas d'apprentissage	Weight Decay	Taille de batch	Époques
Pré-entraînement	2e-6	0,01	4096	1
ChemProt	2e-5	0,01	24	30
PubMedQA	2e-5	0,01	32	5
SNLI	2e-4	0,01	160	10
CoLA	2e-5	0,01	32	10
SNLI	2e-5	0,01	32	5

TABLE 3 – Valeurs des hyperparamètres pour les diverses tâches.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

DACCORD : un jeu de données pour la Détection Automatique d'énonCés COntRaDictoires en français

Maximos Skandalis[1,] Richard Moot[1,] Simon Robillard[1,]
(1) LIRMM, CNRS, Université de Montpellier, 34095 Montpellier, France
[prénom].[nom]@lirmm.fr

RÉSUMÉ

La tâche de détection automatique de contradictions logiques entre énoncés en TALN est une tâche de classification binaire, où chaque paire de phrases reçoit une étiquette selon que les deux phrases se contredisent ou non. Elle peut être utilisée afin de lutter contre la désinformation. Dans cet article, nous présentons DACCORD, un jeu de données dédié à la tâche de détection automatique de contradictions entre phrases en français. Le jeu de données élaboré est actuellement composé de 1034 paires de phrases. Il couvre les thématiques de l'invasion de la Russie en Ukraine en 2022, de la pandémie de Covid-19 et de la crise climatique. Pour mettre en avant les possibilités de notre jeu de données, nous évaluons les performances de certains modèles de transformeurs sur lui. Nous constatons qu'il constitue pour eux un défi plus élevé que les jeux de données existants pour le français, qui sont déjà peu nombreux.

ABSTRACT

DACCORD : a Dataset for Automated deteCtion of COntRaDictions between sentences in French

In NLP, the automatic detection of logical contradictions between statements is a binary classification task, in which a pair of sentences receives a label according to whether or not the two sentences contradict each other. This task has many potential applications, including combating disinformation. In this article, we present DACCORD, a new dataset dedicated to the task of automatically detecting contradictions between sentences in French. The dataset is currently composed of 1034 sentence pairs. It covers the themes of Russia's invasion of Ukraine in 2022, the Covid-19 pandemic, and the climate crisis. To highlight the possibilities of our dataset, we evaluate the performance of some recent Transformer models on it. We conclude that our dataset is considerably more challenging than the few existing datasets for French.

MOTS-CLÉS : détection automatique de contradictions, jeu de données, construction de corpus, tâche de paire de phrases, classification binaire, analyse sémantique de phrases, français.

KEYWORDS: automatic contradiction detection, dataset, corpus construction, sentence pair task, binary classification, semantic analysis of sentences, French.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 285–297.

1 Introduction

1.1 Contexte

Les désinformations (*fake news*) sur les réseaux sociaux sont, de nos jours, un enjeu sociétal majeur. Dans plusieurs pays, y compris la France, diverses entités (individus, états, entreprises) ont mené des campagnes de désinformation pour tenter de manipuler l'opinion publique, la législation, ou le résultat d'élections. Pendant la crise du Covid-19, puis la guerre en Ukraine, ce problème est devenu encore plus apparent. La création relativement récente d'agences comme East StratCom en Europe et Viginum en France montre l'intérêt politique de la lutte contre les fausses informations. De même, des journalistes élaborent des sites de vérification des faits (*fact-checking*) dans plusieurs langues, dont le français.

Pour détecter les *fake news*, plusieurs approches ont été proposées, parmi elles la vérification de rumeurs, la détection de techniques de propagande, l'analyse de sentiment et l'analyse de fiabilité des sources des phrases. Par contre, l'aspect sémantique, bien qu'essentiel pour parler de la véracité des propos, a été peu exploré dans ce cadre précis, surtout pour le français.

1.2 Définition de la tâche

La détection automatique de la relation logique de contradiction constitue une étape nécessaire pour identifier automatiquement des fausses informations. La plupart des fausses informations circulant contredisent un savoir scientifique ou une connaissance du monde établie ou largement acceptée. Par ailleurs, c'est dans la nature de la connaissance scientifique d'être réfutable.

La définition la plus courante des contradictions en langage naturel est que deux énoncés sont contradictoires s'il est impossible pour eux d'être tous les deux vrais dans la même situation (c.-à-d. dans le même modèle en logique). La détection automatique de contradictions entre énoncés consiste donc à indiquer si deux phrases données se contredisent ou non. Dans les données textuelles du monde réel néanmoins, il y a peu de chance de trouver, pour une phrase A, explicitement A et non-A, c'est pour cela que la tâche nécessite une analyse sémantique profonde des phrases confrontées, qui ne sont d'habitude pas identiques entre elles.

L'exemple 1.2.1 donne deux phrases qui sont contradictoires.

(1.2.1)
P : 58 caisses étaient en service dans le centre commercial Amstor le jour de l'attaque, enregistrant ce jour-là un chiffre d'affaires de 2,9 millions de hryvnia ukrainiennes, soit environ 97.000 euros. Des employés du centre commercial, blessés le 27 juin, ont témoigné auprès de l'AFP après l'attaque.
H : Amstor était fermé et vide au moment des frappes par les missiles russes.
R : Contradiction

Les caisses d'un magasin sont en service et produisent un chiffre d'affaires non-nul quand le magasin est ouvert et, sûrement, non-vide. La contradiction ici ne provient pas simplement de l'introduction d'une négation dans l'une des deux phrases. Il s'agit d'un cas difficile à détecter par une machine aujourd'hui, et il illustre l'importance de l'analyse sémantique des phrases pour cette tâche.

En elle-même, la tâche de détection d'énoncés contradictoires ne dit rien de la véracité des énoncés. En revanche, en comparant des textes suspects à une base de connaissances de confiance (par exemple,

réseaux sémantiques, archives d'agences de vérification croisée de faits, sites officiels d'organismes), elle peut être utilisée pour détecter des désinformations. Une comparaison, donc, à une base de connaissances pertinentes (avec la liste actualisée des membres de l'OMS, par exemple) devrait permettre d'établir quelle phrase, entre l'hypothèse et la prémisse, est celle qui est vraie, dans l'exemple 1.2.2.

(1.2.2) P : Madagascar reste toujours membre de l'OMS en mai 2022.
H : Madagascar a quitté, en mai 2022, l'OMS en raison d'un scandale lié au Covid-19.
R : Contradiction

Mais l'intérêt pour la détection automatique de contradictions ne se limite pas seulement à la détection de *fake news*. Pouvoir repérer automatiquement les contradictions dans des données textuelles relève aussi de la compréhension du langage naturel par les machines et du raisonnement automatique sur le langage naturel.

La détection automatique de contradictions textuelles pourrait aussi avoir une place déterminante dans des tâches combinant des données de différentes formes. L'exemple 1.2.3, qui pourrait être une description ou légende d'une vidéo, peut illustrer cette perspective pour une éventuelle tâche multi-modale. Par ailleurs, la diffusion de désinformations dans le monde numérique peut prendre plusieurs formes.

(1.2.3) P : Le camion-remorque de la vidéo transporte un long tube cylindrique, qui est une pièce destinée à une raffinerie de pétrole en Ouzbékistan.
H : Le camion-remorque de la vidéo transporte un missile nucléaire russe.
R : Contradiction

Dans le domaine du traitement automatique du langage naturel (TALN) écrit, les premiers travaux sur la typologie des contradictions (Harabagiu *et al.*, 2006) se concentraient plutôt sur l'usage de la négation et de paraphrases avec des antonymes.

de Marneffe *et al.* (2008) ont élaboré une classification plus détaillée, en analysant les jeux de données RTE 1, 2 et 3 (Dagan *et al.*, 2006) et en construisant leur propre jeu de données de 131 paires de phrases, toutes des contradictions. Ils ont conclu que deux catégories principales de contradictions, avec des sous-catégories, y apparaissent : (1) des contradictions qui se produisent via l'antonymie, la négation et l'incompatibilité date/nombre, qui sont relativement simples à détecter, et (2) des contradictions résultant de l'utilisation de mots factifs ou modaux, de contrastes structuraux et lexicaux subtils, ainsi que de la connaissance du monde.

Ritter *et al.* (2008) ont ajouté à la classification de de Marneffe *et al.* (2008) une sous-catégorie de contradictions sur la connaissance du monde, à savoir des contradictions qui découlent de relations fonctionnelles telles que la relation "x est né à y". Enfin, Tsytsarau *et al.* (2011; 2011) ont étudié les contradictions d'opinions et de sentiments, qu'ils ont divisées en contradictions synchrones et asynchrones (qui correspondent à un changement de sentiment).

1.3 Problématique et motivation

Le but du présent article est de présenter un nouveau jeu de données pour la tâche de détection automatique de contradictions logiques entre énoncés en français. La définition précédente de la contradiction

en langage naturel convient à ce but, car nous nous attachons ici à des cas de contradictions factuelles, qui satisfont cette définition.

La création de ce nouveau jeu de données est motivée par le fait qu'à l'époque où l'usage des modèles de transformeurs (Vaswani *et al.*, 2017) est devenu dominant au sein de la communauté de TALN, un grand besoin pour une quantité importante de jeux de données a émergé.

Un des jeux de données utilisable pour la détection de contradictions entre énoncés en langue française est FraCaS (Amblard *et al.*, 2020). Ce jeu, étant centré sur la tâche d'inférence textuelle, ne contient que 9% d'énoncés contradictoires. De surcroît, FraCaS est un jeu de données relativement petit (346 paires de phrases), par rapport à d'autres jeux de données (toujours) d'inférence textuelle disponibles en anglais. Le plus grand corpus pour la tâche d'inférence textuelle est XNLI (Conneau *et al.*, 2018), mais la plus grande partie (plus de 98%) de sa version française n'est issue que d'une traduction automatique de sa version originale en anglais.

D'autres jeux de données sur l'inférence textuelle ont parfois une vision qui n'est pas purement logique de ce qui constitue une contradiction. Ces jeux peuvent voir leur tâche comme un calcul de score de similarité sémantique entre deux phrases. Cette approche nous semble inadaptée pour la détection de contradictions, car elle peut aboutir à des situations où deux phrases qui ne parlent pas du même sujet sont considérées comme contradictoires.

D'autres jeux de données parus récemment sur la désinformation (Li *et al.*, 2020; Kochkina *et al.*, 2018) sont composés de messages issus de médias sociaux tels que Twitter. Les messages de Twitter, néanmoins, sont en général d'une qualité variable, avec des phrases typiquement courtes et beaucoup d'erreurs et de bruit. De plus, la détection de contradictions sur des messages de Twitter s'effectue très souvent par analyse de sentiment. Par contre, les articles de presse, qui nous intéressent davantage, comprennent des structures linguistiques plus riches et plus complexes. D'autre part, nous travaillons sur des contradictions logiques, donc l'analyse de sentiment n'est pas la voie à prendre.

L'article suit le plan suivant : après cette définition, dans l'introduction, des contradictions et de la tâche de détection automatique de contradictions, la Section 2 présente les jeux de données uni-lingues (anglais ou français) ou bien multi-lingues, qui sont disponibles, au moment de la publication de l'article, pour la tâche en question. Elle aborde aussi les différences entre la tâche de détection de contradictions et la tâche d'inférence textuelle. La Section 3 décrit la démarche suivie pour établir notre nouveau jeu de données sur la Détection Automatique d'énonCés COntRaDictoires en français. Dans la Section 4, nous évaluons certains récents modèles d'apprentissage profond sur le jeu de données construit.

2 État de l'art

2.1 Jeux de données disponibles en anglais

Il existe en anglais plusieurs jeux de données sur la tâche d'inférence textuelle, à savoir RTE (Dagan *et al.*, 2006; Dzikovska *et al.*, 2013) [1], SICK (Marelli *et al.*, 2014), SNLI (Bowman *et al.*, 2015),

1. Nous ne parlons pas ici de la version de RTE intégrée dans GLUE (Wang *et al.*, 2018), où RTE est divisé en deux catégories (inférence, pas d'inférence) et non en trois (inférence, contradiction, ni l'un ni l'autre).

MultiNLI (Williams *et al.*, 2018), XNLI (Conneau *et al.*, 2018) qui est multi-lingue, FraCaS (Cooper *et al.*, 1996).

Ce dernier, FraCaS, contient 346 problèmes d'inférence à traiter. SICK, quant à lui, contient 9840 exemples, dont 14% sont des contradictions. Leur annotation a été faite par *crowdsourcing* et il s'agit de phrases simplifiées en anglais, similaires à celles de FraCaS, comme le notent ses auteurs par ailleurs.

Ensuite, il existe huit corpora RTE, au même nombre que les compétitions homonymes organisées de 2006 à 2013. RTE-1 contient au total 1367 paires de phrases (800 dans le sous-ensemble de test, 287 dans le sous-ensemble dev_1 et 280 dans le sous-ensemble dev_2), dont 252 sont contradictoires (18,43% du corpus), 682 sont des inférences et 433 des cas neutres. RTE-2 est composé de 1600 paires (800 pour le sous-ensemble de test et 800 pour le sous-ensemble dev), dont 215 des contradictions (13,44% des cas), 800 des cas d'inférence et le reste (585) des cas neutres. Enfin, RTE-3 compte également 1600 paires de phrases (800-800), dont 152 des contradictions (9,5%), 822 des cas d'inférence et 626 des cas neutres.

Nie *et al.* (2020) ont transformé le jeu de données anglais FEVER, initialement construit par Thorne *et al.* (2018) avec des phrases de Wikipedia modifiées par des annotateurs et comparées à la section d'introduction d'une liste de pages Wikipedia pour une tâche de vérification automatique de faits, à un jeu de données pour l'inférence textuelle (donc, avec une étiquette "inférence", "contradiction" ou "neutre" pour chaque paire de prémisse-hypothèse fixée pour cette version).

D'autres jeux de données en anglais, qui essaient de traiter en particulier le problème des désinformations, existent, souvent basés sur des messages de Twitter, par exemple PHEME (Kochkina *et al.*, 2018). Pour PHEME, des messages de Twitter ont été classifiés par un journaliste comme "rumeurs" ou "non-rumeurs", puis ceux, qui sont des rumeurs, en "vrais", "faux" ou "non-vérifiées". Additionnellement, des réponses à ces *tweets* ont été annotées comme interrogatives, soutenant, niant ou commentant la rumeur.

2.2 Jeux de données disponibles en français

Comme déjà mentionné (1.3), en français, nous disposons d'une traduction française récente de FraCaS (Amblard *et al.*, 2020). FraCaS classe ses exemples dans les catégories suivantes : Quantificateurs Généralisés, Pluriels, Anaphore (nominale), Ellipse, Adjectifs, Comparatifs, Référence temporelle, Verbes, et Attitudes. Le jeu de données est disponible sous forme de question-réponse et sous forme de prémisse(s)-hypothèse.

XNLI (Conneau *et al.*, 2018) contient des traductions manuelles de l'anglais au français des sous-ensembles de validation et de test de MNLI initial. En particulier, Son sous-ensemble de validation est composé de 2490 paires de phrases (830 contradictions, 830 cas neutres, 830 inférences), alors que le sous-ensemble de test contient 5010 paires (1670 contradictions, 1670 cas neutres, 1670 inférences). Par contre, le sous-ensemble d'entraînement de XNLI en français n'est qu'une traduction automatique (par Conneau *et al.* (2018) et par Hu *et al.* (2020)) de celui de MNLI.

Enfin, MM-COVID (Li *et al.*, 2020) est un jeu de données multi-modal et multi-lingue, qui inclut aussi des exemples en français, mais qui sont tous des messages issus de médias sociaux. Pour ce qui est de son contenu en français, il inclut 2821 tweets dits "faux" (et 4459 réponses), contre 166 tweets dits "vrais" (et 5095 réponses).

2.3 Inférence textuelle et détection des contradictions

La détection d'énoncés contradictoires, avec comme finalité la détection de désinformations, et la tâche d'inférence textuelle, bien qu'elles partagent certains points communs, ne sont pas les mêmes.

D'un autre côté, il est important de faire la distinction entre la détection d'énoncés contradictoires et la détection de désinformations en tant que tâche de détection de rumeurs (Gorrell *et al.*, 2019) ou de techniques de propagande (Da San Martino *et al.*, 2020). De même, nous ne jugeons pas, dans le cadre de notre jeu de données, les sources des phrases ni ne caractérisons leur fiabilité en les séparant en sources fiables et sources non-fiables, comme le font Guibon *et al.* (2019). Nous nous limitons aux événements décrits par les phrases elles-mêmes, car le but est de détecter des contradictions logiques. Ceci dit, le corpus contient des phrases du monde réel, ce qui signifie aussi que, lors de la détection de contradictions logiques, il peut y avoir (et il y en a) des prémisses cachées.

Notre jeu de données sur la détection de contradictions se rapproche plutôt de jeux de données construits pour la tâche d'inférence textuelle (*textual entailment*). Cependant, il existe une différence entre la détection de contradictions et l'inférence textuelle, à savoir que la première est une tâche de classification binaire (les phrases sont contradictoires ou non), alors que la dernière est souvent vue comme une tâche de classification multi-classe ("oui/inférence", "non/contradiction" ou "inconnu/neutre" [2]). Dans notre cas, nous avons choisi de nommer nos deux classes "contradiction" et "compatibles".

Un autre point important est que, par comparaison à la tâche de détection de contradictions, les contradictions sont sous-représentées dans les jeux de données construits pour la tâche d'inférence textuelle. Pour la tâche d'inférence textuelle à trois étiquettes, les contradictions ne constituent qu'un tiers (ou même moins) des exemples dans les jeux de données dédiés. Dans FraCaS par exemple, les contradictions représentent 9% de l'ensemble du corpus, 52% des énoncés sont des inférences, 27% sont des cas neutres, et 12% des exemples nécessitent une réponse plus détaillée.

La raison pour laquelle nous avons choisi d'établir ce corpus sur la relation de contradiction et non pas sur l'inférence textuelle plus généralement est que la contradiction est symétrique. Il est important de voir si des modèles neuronaux sont capables de percevoir et prédire cette symétrie. Ainsi, nous avons intégré dans le corpus quelques exemples qui sont similaires mais inversés, c'est-à-dire que la prémisse prend la place de l'hypothèse et l'hypothèse celle de la prémisse, par rapport à l'exemple précédent. L'inférence textuelle n'est quant à elle pas symétrique, d'où aussi une différence entre les tâches de classification contradiction/pas de contradiction et inférence/pas d'inférence.

Toutefois, ces différences n'empêchent pas que nos résultats puissent s'intégrer avec des approches plus étendues visant des tâches telles que l'inférence textuelle (en divisant notre catégorie "compatibles" en "inférence" et "neutre" par exemple, ou en fusionnant ces deux catégories des autres jeux de données, même si cela aboutirait à des catégories qui ne sont pas équilibrés, ou bien en combinant notre jeu de données avec d'autres jeux).

2. L'hypothèse n'est pas conséquence de la prémisse, mais elle ne la contredit pas non plus.

3 DACCORD, un nouveau jeu de données pour la détection de contradictions

3.1 Méthode de construction du jeu de données

Notre jeu de données a été construit à partir d'articles sur le site factuel.afp.com. AFP Factuel est un site français de vérification de faits par Agence France-Presse. Les paires de phrases ont été manuellement sélectionnées en lisant les articles du site susmentionné et en gardant les phrases qui nous paraissaient d'intérêt pour la tâche étudiée.

Le jeu de données couvre actuellement trois thématiques : l'invasion russe en Ukraine, la pandémie de Covid-19 et la crise climatique. À notre connaissance, c'est en général le tout premier jeu de données de TALN couvrant le conflit entre Russie et Ukraine.

Le jeu de données est composé de 1034 paires de phrases, dont 515 (49,81%) forment des contradictions. Parmi elles, 472 paires de phrases (215 contradictions) ont été recueillies à partir de 106 articles sur la guerre russo-ukrainienne, datant du 24 février 2022 au 3 novembre 2022 (inclus). 450 paires (251 contradictions) ont été retenues à partir de 164 articles publiés du 20 octobre 2021 au 23 novembre 2022 sur la pandémie de Covid-19. Enfin, les 112 paires de phrases (49 contradictions) sur le réchauffement climatique sont issues de 33 articles datant du 16 juillet 2021 au 24 octobre 2022.

3.2 Propriétés du jeu de données

Nous avons choisi AFP Factuel comme source pour recueillir des phrases pour le jeu de données, cependant nous ne nous positionnons pas par rapport à la vérité des énoncés choisis, les thématiques traitées étant délicates et la notion de vérité n'étant formellement pas triviale. Il est indiqué dans le corpus quand une paire de phrases forme une contradiction ou quand les deux phrases d'un exemple sont inter-compatibles, mais il n'est pas indiqué lequel parmi les énoncés est vrai et lequel ne l'est pas.

De plus, chaque paire de phrases est un monde clos. Tout le contexte (par exemple, dates et/ou lieu) se trouve dans ces mêmes phrases. Par contre, il est clair, en regardant les différentes paires, qu'elles peuvent contenir beaucoup de prémisses cachées, qui devront être prises en compte pour appliquer des approches purement formelles/logiques sur le jeu de données.

Les exemples 3.2.1 et 3.2.2, extraits de la partie du corpus sur la pandémie de Covid-19, seraient des exemples de contradiction structurelle d'après la typologie de de Marneffe *et al.* (2008).

(3.2.1) P : De nombreux vaccins utilisés aujourd'hui n'induisent, comme ceux contre le Covid-19, qu'une immunité effective. Le vaccin contre la variole est, quant à lui, un exemple d'immunité "stérilisante".
H : Les vaccins contre le Covid-19 sont un exemple d'immunité stérilisante.
R : Contradiction

(3.2.2) P : Interrogée par l'AFP, l'Autorité régionale de santé (ARS) de Guadeloupe déplore une fausse information circulant et précise que ce n'est jamais elle qui passe les commandes de médicaments.
H : C'est une fausse information que ce n'est pas l'Autorité régionale de santé (ARS) de Guadeloupe qui passe les commandes des médicaments.

R : Contradiction

La contradiction dans 3.2.1 résulte des expressions “ne... que” et “quant à lui” de la prémisse, mais aussi d’une confusion concernant le groupe nominal qui sert de sujet du groupe verbal “être un exemple d’immunité stérilisante”. 3.2.2 pourrait être considéré comme un exemple méta-référentiel de fausse information.

La paire de phrases 3.2.3, qui provient du sous-ensemble sur le conflit ukrainien-russe, est un cas difficile de contradiction numérique.

(3.2.3) P : Cent-trente neuf pays sur les 193 membres de l’Assemblée générale des Nations Unies ont voté contre une résolution demandant à la Russie d’arrêter l’opération d’invasion de l’Ukraine et de retirer l’armée du territoire.
H : Toutes les résolutions présentées devant l’Assemblée générale au sujet de l’Ukraine ont été approuvées par au moins deux tiers des membres présents et votants conformément à la Charte des Nations Unies, et ont, donc, été adoptées par l’Assemblée générale.
R : Contradiction

Il s’agit d’un cas difficile car il n’est pas direct. Il faudrait que la machine arrive à faire le raisonnement que l’expression “cent-trente neuf pays sur les 193” correspond à 72%, et donc que, selon la prémisse, 28% des pays ont voté pour la résolution, alors que l’hypothèse parle d’“au moins 66%” qui votaient pour. Nous pourrions aussi voir cet exemple comme un cas d’antonymie contradictoire, entre les verbes “voté contre” et “approuvées” dans la prémisse et l’hypothèse, respectivement.

La paire de phrases 3.2.4 est donnée à titre d’exemple non-contradictoire.

(3.2.4) P : Le calcul du total des factures d’énergie n’est pas uniquement fondé sur les prix de gros des marchés, et diffère pour les consommateurs britanniques et français, la majorité de ces derniers bénéficiant des tarifs réglementés, dont la hausse a été plafonnée par le gouvernement jusqu’à fin 2022.
H : Il faut bien faire la différence entre les prix de marché et les tarifs réglementés, qui sont fixés par les autorités et qui concernent la plupart des particuliers consommateurs d’électricité en France.
R : Compatibles

Selon le tokéniseur de NLTK (Bird *et al.*, 2009), l’ensemble du jeu de données contient 63326 *tokens* au total (y incluse la ponctuation). La prémisse la plus courte du jeu contient 9 *tokens* (la ponctuation toujours incluse) et se trouve dans le sous-ensemble sur la pandémie de Covid-19, alors que la prémisse la plus longue est de 156 *tokens*, dans la partie sur la guerre entre Russie et Ukraine. Concernant les hypothèses, la plus courte fait partie du sous-ensemble sur le Covid-19 et contient 6 *tokens*, tandis que l’hypothèse la plus longue est composée de 111 *tokens* et porte de nouveau sur la guerre russo-ukrainienne.

Le Tableau 1 donne des détails sur le nombre de *tokens* des phrases par thématique pour DACCORD, ainsi que pour XNLI et FraCaS, à titre comparatif.

Jeux de données		Prémisse plus courte	Prémisse plus longue	Moyenne par prémisse	Somme de *tokens* des prémisses	Hypothèse plus courte	Hypothèse plus longue	Moyenne par hypothèse	Somme de *tokens* des hypothèses
DACCORD	Climat	20	112	50,13	5.614	9	111	43,61	4.884
	Covid-19	9	100	30,29	13.631	6	76	22,25	10.014
	Guerre Rus-Ukr	10	156	34,86	16.455	5	147	26,98	12.734
XNLI (test et val)		2	59	22,55	169.092	3	46	11,66	87.485
FraCaS		2	28	9,13	4.805	4	41	9,49	3.245

TABLE 1 – Nombre de *tokens* dans DACCORD, XNLI et FraCaS

4 Expériences

4.1 Protocole expérimental

Afin d'évaluer la performance de l'état de l'art sur notre nouveau jeu de données, nous avons choisi d'utiliser des modèles d'apprentissage profond basés sur l'architecture de transformeur (Vaswani *et al.*, 2017). Les modèles retenus pour l'évaluation sur le jeu de données DACCORD sont DistilmBERT (Sanh *et al.*, 2019), XLM-R (Conneau *et al.*, 2020), mDeBERTa-v3 (He *et al.*, 2021), et CamemBERT (Martin *et al.*, 2020). Ils sont tous entraînés en partie (DistilmBERT, XLM-R et mDeBERTa) ou entièrement (CamemBERT) sur des données françaises. Pour leur évaluation, nous avons utilisé des versions ajustées à XNLI, disponibles sur huggingface.co.

4.2 Résultats

Le Tableau 2 présente les résultats des expériences menées sur DACCORD. À titre de comparaison, des résultats calculés sur XNLI y sont aussi indiqués. Les modèles mentionnés et accessibles par hyperliens dans le Tableau 2, même quand évalués sur DACCORD, sont des modèles pour l'instant entraînés sur le sous-ensemble d'entraînement de XNLI. Ces données sont issues d'une traduction automatique en français qui limite leur qualité, mais la quantité de données fournie est nécessaire pour permettre l'entraînement des modèles actuels.

L'étude portant sur la détection de contradictions, nous avons calculé l'*accuracy* et le score F1 [3] sur la probabilité prédite par les modèles que l'étiquette "contradiction" soit vraie.

En regardant les résultats, nous constatons, d'abord, une évolution progressive et constante des performances des modèles multi-lingues même sur les tâches uni-lingues (par exemple, mDeBERTa par opposition à DistilmBERT).

De plus, les modèles entraînés sur XNLI présentent, sans exception, des performances inférieures sur DACCORD que sur XNLI. Cela ne constitue pas une surprise, puisque DACCORD est construit de sorte à éprouver les capacités des modèles existants.

On peut, toutefois, observer une cohérence entre les performances sur XNLI des modèles étudiés et

3. Pour rappel, la mesure F1 est la moyenne harmonique de la précision et du rappel.

Modèles	DACCORD		XNLI	
	Accuracy	Score F1	*Accuracy*	Score F1
$\text{DistilmBERT}_{\text{Base-cased}}$	63,73	52,59	79,98	68,01
$\text{XLM-R}_{\text{Base}}$	71,57	67,62	87,17	81,14
$\text{CamemBERT}_{\text{Base, 3-class}}$	77,76	76,19	89,64	85,09
$\text{mDeBERTa-v3}_{\text{Base, XNLI}}$	80,75	78,30	90,98	86,39
$\text{mDeBERTa-v3}_{\text{Base, NLI-2mil7}}$	80,95	78,47	90,76	85,89
$\text{XLM-R}_{\text{Large}}$	82,01	80,00	**96,49**	**94,74**
$\text{CamemBERT}_{\text{Large, 3-class}}$	83,27	81,01	92,30	88,12
$\text{CamemBERT}_{\text{Large, 2-class}}$	**84,24**	**82,49**	91,70	87,66

TABLE 2 – Résultats de détection de contradictions par les transformeurs sur DACCORD et XNLI

leurs performances sur DACCORD, les deux meilleurs modèles pour XNLI par exemple (XLM-R et CamemBERT) étant aussi les meilleurs modèles pour DACCORD, même si les résultats sont inférieurs à ceux obtenus sur XNLI.

Pour information, tous les modèles évalués dans l'article ont échoué à détecter la contradiction dans l'exemple 3.2.2, et seulement $\text{mDeBERTa-v3}_{\text{Base, NLI-2mil7}}$ a réussi à trouver la bonne étiquette pour l'exemple 3.2.1. Quant à l'exemple 3.2.3 donné, tous les modèles ont correctement prédit son étiquette, sauf pour $\text{XLM-R}_{\text{Base}}$. Enfin, l'étiquette pour l'exemple 1.2.1 n'a pas été correctement prédit par DistilmBERT et $\text{CamemBERT}_{\text{Large, 3-class}}$.

Nous avons déjà fait remarquer dans 2.3 que les contradictions sont naturellement symétriques. Afin de tester le comportement des modèles d'apprentissage profond vis-à-vis de cette symétrie, nous avons effectué une dernière expérience, en échangeant la place de toutes les prémisses avec celle des hypothèses. Ses résultats sont consultables dans le Tableau 3.

Modèles	DACCORD		XNLI	
	Accuracy	Score F1	*Accuracy*	Score F1
$\text{XLM-R}_{\text{Large}}$	77,47	73,79	**83,77**	**73,12**
$\text{CamemBERT}_{\text{Large, 2-class}}$	**82,50**	**80,22**	81,64	69,72

TABLE 3 – Détection des contradictions avec la place des prémisses et des hypothèses inversée

Dans ce nouveau test, $\text{CamemBERT}_{\text{Large, 2-class}}$ obtient maintenant un score d'*accuracy* de 82,5% et un score F1 de 80,22% sur DACCORD, au lieu de 84,24% et 82,49%, respectivement. De même, il obtient un score d'*accuracy* de 81,64% et un score F1 de 69,72% sur XNLI, au lieu de 91,7% et 87,66%, respectivement. Les résultats de $\text{XLM-R}_{\text{Large}}$ sont aussi en baisse dans ce scénario avec les prémisses et les hypothèses échangées : sur XNLI, son score d'*accuracy* devient 83,77% et son score F1 73,12%, par opposition à 96,49% et 94,74%, respectivement, avant. Enfin, sur DACCORD, son *accuracy* est en baisse à 77,47% et son score F1 à 73,79%. Ces derniers résultats pourraient suggérer que cet aspect de la relation de contradiction n'est pas suffisamment pris en compte par les jeux de données existants utilisés largement pour l'entraînement des modèles.

5 Conclusion et perspectives

Dans cet article, nous avons présenté DACCORD, un nouveau jeu de données pour la tâche de détection automatique d'énoncés contradictoires en français. Il est composé de 1034 paires de phrases, toutes récupérées manuellement, dont 515 contradictions. À notre connaissance, c'est le premier corpus en français exclusivement dédié à la tâche de détection de contradictions et couvrant les thématiques du Covid-19 et de la guerre entre Russie et Ukraine. De plus, c'est le jeu de données le plus compliqué pour cette tâche, étant données la longueur des prémisses et des hypothèses incluses et la nature des sources utilisées (articles de presse et non messages de médias sociaux). Lors de l'évaluation des modèles examinés, DACCORD s'avère être plus difficile pour eux que le jeu de données sur l'inférence textuelle XNLI.

Nous comptons par la suite expérimenter avec des méthodes neuro-symboliques sur le corpus construit. Nous souhaiterions, enfin, enrichir le corpus avec davantage de phrases et de thématiques de l'actualité mais peu incorporées dans les jeux de données disponibles.

Remerciements

La recherche présentée a été réalisée avec le soutien financier du Ministère des Armées – Agence de l'innovation de défense (AID), que nous en remercions. Ce travail a également bénéficié du soutien de l'ICO, Institut Cybersécurité d'Occitanie, financé par la Région Occitanie, France, auquel nous exprimons aussi notre gratitude.

Références

AMBLARD M., BEYSSON C., DE GROOTE P., GUILLAUME B. & POGODALLA S. (2020). A French version of the FraCaS test suite. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 5887–5895, Marseille, France : European Language Resources Association.

BIRD S., KLEIN E. & LOPER E. (2009). *Natural language processing with Python : analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

BOWMAN S. R., ANGELI G., POTTS C. & MANNING C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 632–642, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1075.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8440–8451, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.747.

CONNEAU A., RINOTT R., LAMPLE G., WILLIAMS A., BOWMAN S. R., SCHWENK H. & STOYANOV V. (2018). Xnli : Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* : Association for Computational Linguistics.

COOPER R., CROUCH R., VAN EIJCK J., FOX C., VAN GENABITH J., JASPARS J., KAMP H., PINKAL M., MILWARD D., POESIO M., PULMAN S., BRISCOE T., MAIER H. & KONRAD K. (1996). *Using the Framework*. Rapport interne, FraCaS : A Framework for Computational Semantics. FraCaS deliverable D16, 136 pages, also available by anonymous ftp from ftp://ftp.cogsci.ed.ac.uk/pub/FRACAS/del16.ps.gz.

DA SAN MARTINO G., BARRÓN-CEDEÑO A., WACHSMUTH H., PETROV R. & NAKOV P. (2020). SemEval-2020 task 11 : Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, p. 1377–1414, Barcelona (online) : International Committee for Computational Linguistics. DOI : 10.18653/v1/2020.semeval-1.186.

DAGAN I., GLICKMAN O. & MAGNINI B. (2006). The pascal recognising textual entailment challenge. In J. QUIÑONERO-CANDELA, I. DAGAN, B. MAGNINI & F. D'ALCHÉ BUC, Éds., *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, p. 177–190, Berlin, Heidelberg : Springer Berlin Heidelberg.

DE MARNEFFE M.-C., RAFFERTY A. N. & MANNING C. D. (2008). Finding contradictions in text. In *Proceedings of ACL-08 : HLT*, p. 1039–1047, Columbus, Ohio : Association for Computational Linguistics.

DZIKOVSKA M., NIELSEN R., BREW C., LEACOCK C., GIAMPICCOLO D., BENTIVOGLI L., CLARK P., DAGAN I. & DANG H. T. (2013). SemEval-2013 task 7 : The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, p. 263–274, Atlanta, Georgia, USA : Association for Computational Linguistics.

GORRELL G., KOCHKINA E., LIAKATA M., AKER A., ZUBIAGA A., BONTCHEVA K. & DERCZYNSKI L. (2019). SemEval-2019 task 7 : RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, p. 845–854, Minneapolis, Minnesota, USA : Association for Computational Linguistics. DOI : 10.18653/v1/S19-2147.

GUIBON G., ERMAKOVA L., SEFFIH H., FIRSOV A. & LE NOÉ-BIENVENU G. (2019). Multilingual Fake News Detection with Satire. In *CICLing : International Conference on Computational Linguistics and Intelligent Text Processing*, La Rochelle, France. HAL : halshs-02391141.

HARABAGIU S., HICKL A. & LACATUSU F. (2006). Negation, contrast and contradiction in text processing. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, p. 755–762 : AAAI Press.

HE P., GAO J. & CHEN W. (2021). Debertav3 : Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. DOI : 10.48550/ARXIV.2111.09543.

HU J., RUDER S., SIDDHANT A., NEUBIG G., FIRAT O. & JOHNSON M. (2020). XTREME : A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In H. D. III & A. SINGH, Éds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 de *Proceedings of Machine Learning Research*, p. 4411–4421 : PMLR.

KOCHKINA E., LIAKATA M. & ZUBIAGA A. (2018). All-in-one : Multi-task learning for rumour verification. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 3402–3413, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

LI Y., JIANG B., SHU K. & LIU H. (2020). Mm-covid : A multilingual and multimodal data repository for combating covid-19 disinformation.

MARELLI M., MENINI S., BARONI M., BENTIVOGLI L., BERNARDI R. & ZAMPARELLI R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, p. 216–223, Reykjavik, Iceland : European Language Resources Association (ELRA).

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

NIE Y., WILLIAMS A., DINAN E., BANSAL M., WESTON J. & KIELA D. (2020). Adversarial NLI : A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4885–4901, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.441.

RITTER A., SODERLAND S., DOWNEY D. & ETZIONI O. (2008). It's a contradiction – no, it's not : A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, p. 11–20, Honolulu, Hawaii : Association for Computational Linguistics.

SANH V., DEBUT L., CHAUMOND J. & WOLF T. (2019). Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. DOI : 10.48550/ARXIV.1910.01108.

THORNE J., VLACHOS A., CHRISTODOULOPOULOS C. & MITTAL A. (2018). FEVER : a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 809–819, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1074.

TSYTSARAU M. & PALPANAS T. (2011). Towards a framework for detecting and managing opinion contradictions. In M. SPILIOPOULOU, H. WANG, D. J. COOK, J. PEI, W. WANG, O. R. ZAÏANE & X. WU, Éds., *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*, p. 1219–1222 : IEEE Computer Society. DOI : 10.1109/ICDMW.2011.167.

TSYTSARAU M., PALPANAS T. & DENECKE K. (2011). Scalable detection of sentiment-based contradictions. *DiversiWeb, WWW*, **1**, 9–16.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 30 : Curran Associates, Inc.

WANG A., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2018). GLUE : A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 353–355, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5446.

WILLIAMS A., NANGIA N. & BOWMAN S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1112–1122 : Association for Computational Linguistics.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Exploitation de plongements de graphes pour l'extraction de relations biomédicales

Anfu TANG[1, 2] Robert Bossy [1] Louise Deléger[1]
Claire Nédellec[1] Pierre Zweigenbaum[2]
(1) Université Paris-Saclay, INRAE, MaIAGE, Jouy-en-Josas, France
(2) Université Paris-Saclay, CNRS, LISN, Orsay, France
anfu.tang@inrae.fr, robert.bossy@inrae.fr, louise.deleger@inrae.fr,
claire.nedellec@inrae.fr, pz@lisn.fr

RÉSUMÉ

L'intégration de connaissances externes dans les modèles neuronaux est très étudiée pour améliorer les performances des modèles de langue pré-entraînés, notamment en domaine biomédical. Dans cet article, nous explorons la contribution de plongements de bases de connaissances à une tâche d'extraction de relations. Pour deux mentions d'entités candidates dans un texte, nous faisons l'hypothèse que la connaissance de relations entre elles, issue d'une base de connaissances (BC) externe, aide à prédire l'existence d'une relation dans le texte, y compris lorsque les relations de BC sont différentes de celles du texte. Notre approche consiste à calculer des plongements du graphe de BC et à estimer la possibilité pour chaque paire d'entité du texte qu'elle soit reliée par une relation de BC. Les expériences menées sur trois tâches d'extraction de relations en domaine biomédical montrent que notre méthode surpasse le modèle PubMedBERT de base et obtient des performances comparables aux méthodes de l'état de l'art.

ABSTRACT

Exploiting Graph Embeddings from Knowledge Bases for Neural Biomedical Relation Extraction

Integrating external knowledge into neural models has been extensively studied to improve the performance of pre-trained Language Models, especially in the biomedical domain. In this paper, we explore the contribution of knowledge base embeddings to relation extraction (RE) tasks. Given two candidate entity mentions in a text, we hypothesize that knowing which relations exist between them in an external knowledge base (KB) helps to predict the existence of a relation in the text, even when the KB relations are different from those of the RE task. Our approach consists in computing KB graph embeddings and in estimating the possibility that each pair of entities is linked by a KB relation. Experiments conducted on three biomedical RE tasks show that our method outperforms the baseline PubMedBERT model and yields comparable performance to state-of-the-art methods.

MOTS-CLÉS : Extraction de relations, BERT, plongements de graphes, base de connaissances.

KEYWORDS: Relation Extraction, BERT, Graph Embedding, Knowledge Base.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 298–310.

1 Introduction

L'extraction de relations est une tâche importante du traitement automatique des langues sur laquelle de nombreuses études existent. Elle consiste à identifier le type de relation entre une paire d'entités étant donné une phrase entière. Un exemple d'extraction de relations est montré dans la figure 1 : l'objectif est de déterminer quelle relation existe entre *Argatroban* et *thrombin*, par exemple *CPR:4*.

FIGURE 1 – Un exemple venant du corpus d'extraction de relations ChemProt (Krallinger *et al.*, 2017). Une relation "CPR:4" est annotée dans la phrase entre *Argatroban* et *thrombin* ; on trouve une relation "decrease^activity" entre ces entités dans la BC externe CTD (Davis *et al.*, 2023).

Les modèles de langue pré-entraînés fondés sur les architectures de type Transformer (Vaswani *et al.*, 2017) tels que BERT (Devlin *et al.*, 2019) obtiennent des performances à l'état de l'art dans diverses tâches de Traitement Automatique des Langues (TAL). Le modèle de langue BERT est pré-entraîné sur un corpus du domaine général, ce qui entraîne des limitations lorsqu'il est appliqué à un domaine spécifique. Pour adapter BERT à des domaines particuliers, des variantes de BERT (Lee *et al.*, 2020; Beltagy *et al.*, 2019; Gu *et al.*, 2021) ont été pré-entraînées sur des corpus de ces domaines, soit en partant de la version pré-entraînée dans le domaine général (SciBERT (Beltagy *et al.*, 2019), BioBERT (Lee *et al.*, 2020)), soit en partant de zéro (PubMedBERT (Gu *et al.*, 2021), CharacterBERT (El Boukkouri *et al.*, 2020)). Dans ces domaines, ces modèles spécifiques surpassent le modèle BERT pré-entraîné sur un corpus du domaine général (Lee *et al.*, 2020).

Le pré-entraînement de BERT sur un corpus de textes n'a pas d'objectif explicite visant à acquérir des connaissances factuelles qui pourraient contribuer à réaliser notre objectif d'extraction de relations. Certains travaux (Wang *et al.*, 2021; Hao *et al.*, 2020) ont proposé d'ajouter un objectif de pré-entraînement lié à une base de connaissances (BC) pour injecter des connaissances factuelles. D'autres (Iinuma *et al.*, 2022; Hao *et al.*, 2020) se concentrent sur l'utilisation de bases de connaissances pour obtenir des données supplémentaires. Dans la communauté du Web sémantique, des méthodes de plongement de graphes telles que (Ribeiro *et al.*, 2017; Bordes *et al.*, 2013; Sun *et al.*, 2019) ont été développées. Elles fournissent des outils pour intégrer des informations provenant de graphes de connaissance.

Cependant, dans un contexte d'extraction de relations à partir de textes, les bases de connaissances de domaines spécifiques contiennent souvent des relations différentes de celles qui sont recherchées dans les textes. Prenons l'exemple de *Comparative Toxicogenomics Database* (CTD) (Davis *et al.*, 2023) et du corpus ChemProt (Krallinger *et al.*, 2017) fréquemment exploités ensemble pour l'extraction de relations d'interactions entre produits chimiques et gènes. Dans CTD il existe des relations pour 134 interactions, telles que "*affects_stability*", "*increase_reaction*", etc., mais dans le corpus ChemProt, seulement 6 interactions sont recherchées, par exemple "*inhibitor*", "*downregulator*", etc. Les relations de CTD sont plus précises que celles de ChemProt et elles ne sont pas directement alignables, ce qui rend leur exploitation plus complexe. Cependant, nous supposons que les relations

de CTD peuvent être utiles ; par exemple, dans la figure 1, la relation *decrease_activity* trouvée dans CTD peut suggérer qu'il est plus probable que la relation à prédire appartienne à la classe *CPR:4*.

Dans cet article, nous proposons KB-PubMedBERT, un modèle neuronal spécifiquement conçu pour l'extraction de relations biomédicales capable d'exploiter des relations de bases externes, notamment lorsqu'elles sont différentes de celles qui sont à extraire des textes. Notre architecture se compose du modèle PubMedBERT pré-entraîné dans le domaine biomédical (Gu *et al.*, 2021) et d'un composant de plongement de graphes basé sur la méthode RotatE (Sun *et al.*, 2019). Nous partons de graphes de connaissances du domaine qui contiennent des relations possiblement liées aux relations cibles des tâches d'extraction de relations : nous faisons l'hypothèse qu'elles peuvent aider à améliorer la classification des relations cibles. Nous utilisons des plongements de ces graphes de connaissances pour estimer la possibilité que des relations de la base de connaissances existent entre deux mentions d'entités. En ajoutant ce profil de possibilités à la sortie du modèle de langue pré-entraîné, nous faisons en sorte que notre modèle encode à la fois les informations textuelles et les informations de la base de connaissances. Nous supposons qu'il devrait ainsi être plus performant dans les tâches d'extraction de relations. À notre connaissance, nous sommes les premiers à étudier comment l'utilisation de relations d'un graphe de connaissances qui ne sont pas les relations cibles peut aider à améliorer l'extraction de relations à l'aide d'un modèle de langue pré-entraîné.

Notre article est organisé comme suit. Nous présentons d'abord RotatE, la méthode de plongement de graphes utilisée comme composant de notre modèle, puis des études antérieures sur l'intégration de connaissances de BC dans des modèles de langues pré-entraînés (section 2). Nous présentons ensuite l'architecture de notre modèle et l'hypothèse sous-jacente (section 3). Nous détaillons enfin les expériences menées et les résultats obtenus (section 4), puis concluons (section 5).

2 Travaux connexes

Dans cette section, nous présentons d'abord des études antérieures sur un composant important de notre modèle : les méthodes de plongement de graphe. Nous présentons ensuite d'autres méthodes visant à intégrer les informations d'une base de connaissances dans les modèles neuronaux. Ces recherches ont en commun un objectif de performance en domaine de spécialité.

2.1 Plongement de graphe

Les méthodes de plongement de graphe apprennent des représentations vectorielles pour les sommets (concepts) et les arêtes (relations) de graphes. Inspirées de word2vec (Mikolov *et al.*, 2013), les méthodes basées sur le contexte telles que node2vec (Grover & Leskovec, 2016) et Struc2vec (Ribeiro *et al.*, 2017) consistent d'abord à échantillonner aléatoirement des séquences de sommets à partir du graphe, puis à utiliser des sommets voisins comme contexte pour apprendre des plongements de sommets. D'autres méthodes telles que TransE (Bordes *et al.*, 2013) et RotatE (Sun *et al.*, 2019) consistent à modéliser une arête possédant une étiquette donnée comme une transformation entre les vecteurs des sommets qu'elle relie. Par exemple, étant donné une arête r entre deux sommets h et t, les vecteurs correspondants sont notés $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, TransE modélise r comme une translation telle que $\mathbf{t} = \mathbf{h} + \mathbf{r}$. Suivant le même principe mais au lieu d'une translation, la méthode RotatE modélise les arêtes comme des rotations dans l'espace vectoriel complexe : $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$, où $\circ$ est le produit de Hadamard (élément par élément). Dans notre travail, nous choisissons la méthode RotatE car elle

surpasse systématiquement TransE sur plusieurs jeux de données, comme indiqué dans (Sun *et al.*, 2019).

2.2 Intégration des informations d'une base de connaissances dans BERT

Nous classons les méthodes existantes en deux types : 1) utilisation de la base de connaissances pour obtenir des données d'entraînement supplémentaires ; 2) modification des objectifs de pré-entraînement de BERT. Les deux types de méthodes peuvent être utilisées ensemble ou séparément.

2.2.1 Ajout de données d'entraînement à l'aide d'une base de connaissances

Les méthodes suivantes sont représentatives de la diversité des approches. Wang *et al.* (2021) proposent de figer les poids d'un BERT pré-entraîné et de pré-entraîner un encodeur supplémentaire à base de Transformer, dit 'adaptateur factuel', sur des phrases collectées à partir de Wikipédia. Cet adaptateur est entraîné à la classification de relations sur un jeu de données obtenu par alignement entre des triplets de Wikidata et des phrases de Wikipédia (Elsahar *et al.*, 2018). Hao *et al.* (2020) proposent plutôt de générer directement des phrases au format "[CLS] concept_1 [relation] concept_2 [SEP]" où (concept_1, relation, concept_2) est un triplet de la base de connaissances ou un exemple négatif, et [CLS] et [SEP] sont des sous-mots utilisés respectivement pour marquer le début et la fin des phrases, comme indiqué dans (Devlin *et al.*, 2019). Ces phrases artificielles sont ensuite utilisées pour pré-entraîner BERT à une tâche de classification de relations. Weber *et al.* (2022) proposent d'ajouter directement les définitions de produits chimiques obtenues à partir d'une base de connaissances comme données supplémentaires pendant l'affinage du modèle.

2.2.2 Mise à jour des objectifs de pré-entraînement

Les objectifs originaux du pré-entraînement de BERT sont la prédiction de mot masqué et la prédiction de la phrase suivante. Dans UmlsBERT, au lieu de masquer aléatoirement un mot et de demander au modèle de langue de le prédire, Michalopoulos *et al.* (2021) proposent de prédire en plus les entités qui partagent le même identifiant unique de concept (CUI) que le mot masqué si ce mot fait partie d'une entité reconnue dans le Metathesaurus de l'UMLS (Bodenreider, 2004). Dans ERNIE, Zhang *et al.* (2019b) ajoutent un troisième objectif de pré-entraînement associé à un liage référentiel entre tokens et concepts de BC : ils masquent aléatoirement le liage d'un token à un concept et demandent au modèle de prédire le concept masqué. Dans KeBioLM, Yuan *et al.* (2021) ajoutent deux tâches dans le pré-entraînement : la détection d'entités et le liage référentiel. Selon les résultats expérimentales, ces méthodes montrent des performances plus élevées pour des tâches de TAL biomédicales que celles sans modification du pré-entraînement de BERT.

3 Méthode proposée

Dans cette partie, nous présentons notre modèle KB-PubMedBERT qui exploite une base de connaissances. Ce modèle contient deux composants : PubMedBERT pré-entraîné, et un composant de plongement de graphe qui inclut une couche de plongement de concepts et une couche de plongement

de relations. Nous choisissons PubMedBERT comme base car il crée son propre vocabulaire de sous-mots contenant des termes biomédicaux, et ses performances dépassent des variantes de BERT spécifiques précédentes telles que BioBERT et SciBERT sur plusieurs tâches biomédicales (Gu *et al.*, 2021).

3.1 Hypothèse

La plupart des modèles antérieurs qui utilisent une base de connaissances (Zhang *et al.*, 2019b; Yuan *et al.*, 2021) se concentrent sur l'intégration d'informations sur les entités de la base de connaissances (BC) dans des modèles de langue pré-entraînés. Cependant, nous soutenons que l'incorporation d'informations sur les *relations* de la BC est également importante, en particulier pour les tâches d'extraction de relations dans des textes. Un défi auquel nous sommes confrontés en utilisant les relations d'une BC est que dans la plupart des cas, ces relations sont différentes de celles des tâches d'extraction de relations. Cependant, dans une BC liée à un domaine spécifique, les relations de la BC ont des chances d'être sémantiquement liées aux relations des textes, même si ce n'est que faiblement. Par conséquent, trouver un moyen d'exprimer la proximité sémantique entre les relations de la BC et les relations des textes est un point crucial dans la construction d'une architecture d'extraction de relations exploitant une BC. Par exemple, Iinuma *et al.* (2022) créent manuellement un alignement entre les relations de la BC et les relations cibles, puis s'en servent pour créer des données de supervision distante. Nous proposons d'aller plus loin en supprimant cette opération coûteuse d'alignement manuel et en faisant en sorte que le modèle neuronal apprenne cette correspondance automatiquement. Nous émettons l'hypothèse que notre modèle neuronal est capable de construire un alignement souple entre les relations de la BC et les relations du texte, et que l'ajout de ces suggestions de relations hypothétiques en sus de l'encodage du texte par PubMedBERT peut améliorer les performances du modèle en extraction de relations.

3.2 Architecture du modèle

La figure 2 présente une vue d'ensemble de l'architecture de notre modèle. Ce modèle prend deux entrées : la phrase s, et les identifiants des concepts des entités source $subj$ et cible obj. Les plongements de concepts et de relations sont respectivement initialisés avec des plongements de concepts et de relations de la BC entraînés à l'aide de RotatE, et les plongements des concepts présents dans le texte mais absents de la BC sont initialisés aléatoirement. Une fois initialisés, les plongements de concepts et de relations sont ajustés pendant l'entraînement du modèle. Le flux de données dans notre modèle est le suivant : pour deux entités candidates du texte source et cible, dont la relation est à prédire, et dont nous disposons du concept associé, nous obtenons d'abord les plongements $\mathbf{e_{subj}}$, $\mathbf{e_{obj}}$ pour la source et la cible, nous calculons ensuite les M scores de la formule (1) :

$$score(r_i) = (\gamma - ||\mathbf{e_{subj}} \circ \mathbf{r_i} - \mathbf{e_{obj}}||)^{\top}_{i=1,2,...,M} \quad (1)$$

où γ est une marge fixe, un hyper-paramètre dont la valeur est fixée pendant l'entraînement de RotatE. Cette définition du score vient de la fonction de coût que l'on utilise pour entraîner les plongements RotatE, comme indiqué dans (Sun *et al.*, 2019). Selon la définition de RotatE, un $score(r_i)$ élevé reflète la possibilité que la relation r_i soit valide entre $subj$ et obj pour la BC. Soit $\mathbf{h}_s$ l'encodage de s par PubMedBERT et $\mathbf{h}_{score}$ un vecteur de dimension M contenant les scores de toutes les relations

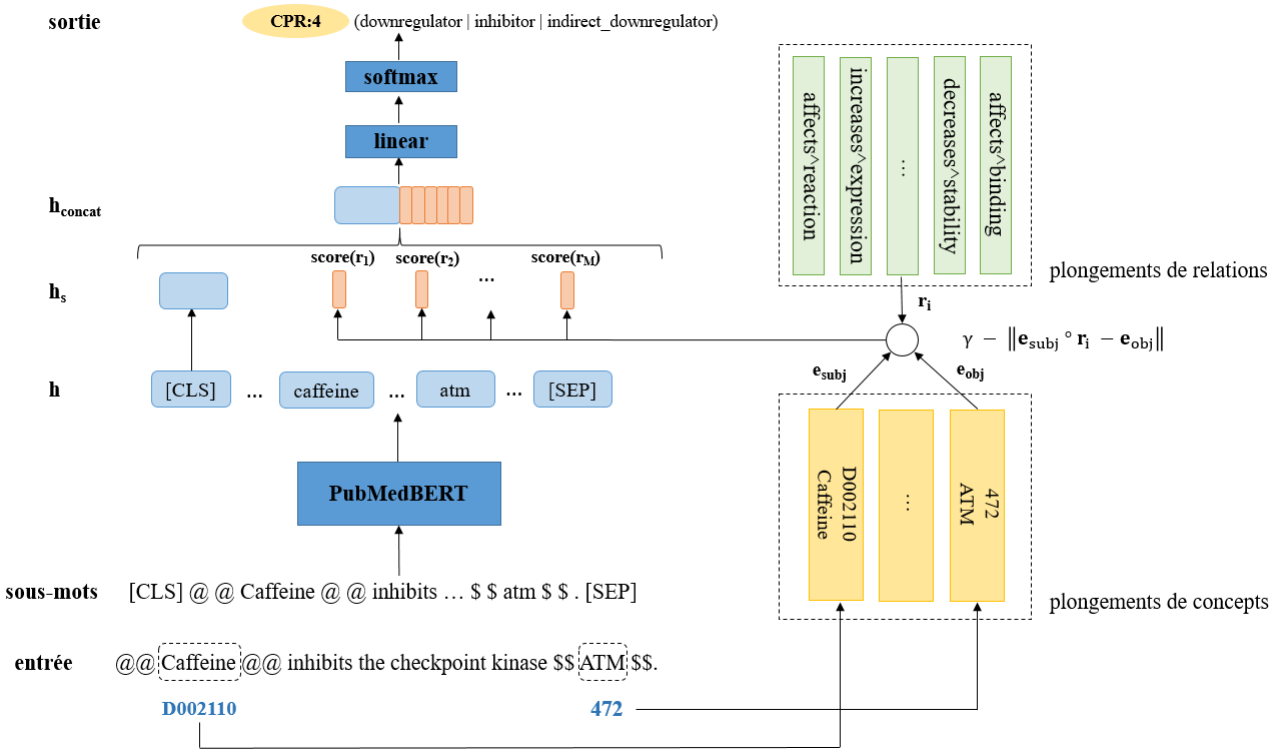


FIGURE 2 – Architecture globale de KB-PubMedBERT.

de BC, nous calculons la représentation combinée selon la formule (2) :

$$\mathbf{h}_{concat} = [\mathbf{h}_s; \mathbf{h}_{score}] \tag{2}$$

où $[.;.]$ désigne la concaténation vectorielle. La représentation combinée est ensuite transmise à une couche entièrement connectée suivie d'une couche softmax qui calcule la possibilité que les relations cibles soient exprimées entre les entités considérées. L'ensemble du modèle est affiné en utilisant l'entropie croisée comme fonction de coût avec des poids de PubMedBERT non-figés.

4 Expérimentations et Résultats

4.1 Jeux de données

Nous évaluons notre modèle sur trois tâches d'extraction de relations biomédicales dans des textes en anglais choisies pour la diversité de leurs caractéristiques. Les statistiques des corpus correspondants sont résumées dans le tableau 1.

1. La tâche ChemProt (Krallinger *et al.*, 2017) porte sur cinq types d'interactions de haut niveau entre produits chimiques et gènes, soit au total 6 relations dont la relation nulle (pas de relation).
2. La tâche DrugProt (Miranda *et al.*, 2021) porte sur 14 types d'interactions entre produits chimiques et gènes, dont la relation nulle.

	ChemProt	DrugProt	BB-Rel$_p$
nb. classes de relations	6	14	2
nb. exemples : entraînement	13 110	64 745	3 016
nb. exemples : développement	8 329	13 399	2 000
nb. exemples : test	10 990	238 694	2 473

TABLE 1 – Extraction de relations biomédicales : jeux de données ChemProt, DrugProt et BB-Rel$_p$.

3. La tâche BB-Rel de Bacteria Biotope (Bossy *et al.*, 2019) porte sur deux relations : "*lives_in*" entre micro-organismes et habitats ou zones géographiques, et "*exhibits*" entre micro-organismes et phénotypes. Comme nous ne disposons que de bases de connaissances portant sur les relations entre micro-organismes et habitats mais pas entre micro-organismes et zone géographique ou phénotypes, nous extrayons du jeu de données complet BB-Rel le sous-ensemble portant sur notre relation cible entre paires d'entités (micro-organisme, habitat). Nous désignons ce sous-ensemble par BB-Rel$_p$ (p pour *partiel*).

Nous choisissons des bases de connaissances adaptées à ces différentes tâches. Pour les tâches ChemProt et DrugProt, nous choisissons CTD (Davis *et al.*, 2023) qui recense 134 types d'interactions entre produits chimiques et gènes, comme "*affects^reaction*" ou "*increases^stability*". Pour obtenir une base de connaissances pour la tâche BB-rel$_p$, nous extrayons de la base de connaissances Omnicrobe (Dérozier *et al.*, 2023) les entités normalisées et la relation "*lives_in*" entre microorgansismes et habitats provenant des sources de référence, BacDive, CIRM et GenBank.

4.2 Cadre expérimental

Pré-traitement Pour chacun des trois jeux de données, nous considérons uniquement les relations intra-phrase [1]. Comme la plupart des études précédentes sur l'extraction des relations telles que (Lee *et al.*, 2020; Gu *et al.*, 2021), nous utilisons deux types de marqueurs pour ajouter des informations de position sur les arguments d'une relation candidate : "@@" au début et à la fin de la mention de l'entité source ; "$$" au début et à la fin de la mention de l'entité cible. L'objectif de ces marqueurs est de fournir au modèle l'information de position des entités ciblées.

Liage référentiel. Pour une tâche d'extraction de relations, l'alignement entre les entités des textes et les concepts de la BC n'est pas toujours donné. Dans nos expériences, cet alignement est soit fourni comme référence dans les corpus annotés, soit obtenu par des outils existants pré-entraînés pour le liage référentiel (normalisation d'entités). La façon dont nous obtenons les normalisations des entités est résumée dans le tableau 2. Pour ChemProt et DrugProt, nous effectuons le liage référentiel des mentions d'entités vers des concepts CTD à l'aide de la méthode BioSyn (Sung *et al.*, 2020) [2]. Pour BB-Rel$_p$, les entités de type microbe sont normalisées par la taxonomie des espèces du NCBI à l'aide du modèle proposé par le meilleur participant (Mao & Liu, 2019) [3] à la tâche BB-Norm (Bossy *et al.*, 2019), et les entités de type habitat sont normalisées par l'ontologie OntoBiotope à

1. L'évaluation classe en faux-négatifs toutes les relations inter-phrases.
2. Nous utilisons deux modèles BioSyn entraînés par Sung *et al.* (2020) : biosyn-sapbert-bc5cdr-chemical pour les produits chimiques, et biosyn-sapbert-bc2gn pour les gènes. Les deux modèles obtiennent respectivement 96,6 et 91,3 comme acc@1 sur les tâches de normalisation d'entités correspondantes.
3. Ce modèle obtient 0,78 comme précision pour la normalisation des microbes de la tâche BB-Norm.

	ChemProt & DrugProt	BB-Rel$_p$
jeu d'entraînement	BioSyn	*gold*
jeu de validation	BioSyn	*gold*
jeu de test	BioSyn	C-Norm, *regression*

TABLE 2 – Sources de normalisation des entités de ChemProt, DrugProt et BB-rel$_p$: respectivement sur le jeu d'entraînement, de validation et de test. "*gold*" représente les annotations manuelles fournies dans BB-Norm ; "*regression*" réfère au modèle de (Mao & Liu, 2019) qui est un modèle de régression.

l'aide de la méthode état de l'art C-Norm (Ferré *et al.*, 2020) [4]. Soulignons que même si sur chaque tâche, les entités textuelles sont normalisées avec les concepts des référentiels utilisés par la BC choisie, il se peut que certains concepts d'entités n'aient pas de plongements pré-calculés par RotatE. En effet, il existe dans les BC des concepts isolés, i.e., des concepts n'ayant pas de relations avec d'autres concepts, ils ne sont alors pas utilisés pour entraîner RotatE. Pour les entités correspondant à des concepts isolés, nous utilisons un vecteur aléatoirement initialisé au début de l'affinage de KB-PubMedBERT.

Base de comparaison. Nous utilisons le modèle pré-entraîné PubMedBERT comme base de comparaison, car c'est le modèle dont est dérivé notre méthode. Sur chaque jeu de données, PubMedBERT est affiné pour classifier les relations cibles de chaque tâche. La comparaison des performances de notre modèle KB-PubMedBERT à celles de cette architecture de base permet de montrer directement si les informations intégrées à partir de la BC sont utiles.

De manière classique, le vecteur du token [CLS] encode la phrase. On le fait passer à travers une couche linéaire, puis une fonction SoftMax pour obtenir les probabilités des relations à prédire.

Hyperparamètres. Nous utilisons l'implémentation officielle [5] de RotatE (Sun *et al.*, 2019) pour calculer les plongements de concepts et de relations de la BC. Nous fixons expérimentalement la dimension des plongements à 200, γ à 24,0 et le taux d'apprentissage à $1e^{-4}$. Pour chaque jeu de données d'extraction de relations, nous utilisons l'ensemble de développement pour optimiser les hyperparamètres. Nous effectuons une recherche en grille pour deux hyperparamètres : le taux d'apprentissage ($1e^{-5}, 2e^{-5}, 5e^{-5}$) et la taille de lot $(8, 16)$. Nous maintenons le taux d'apprentissage constant pendant l'affinage et fixons le nombre d'époques à 15. Pour la base de comparaison, comme pour KB-PubMedBERT, chaque expérience est répétée avec 5 amorces différentes.

Évaluation. Le score Micro F1 excluant la relation nulle [6] est la métrique d'évaluation standard pour les trois jeux de données : concrètement, pour calculer le score F1, les vrais positifs pris en compte sont uniquement les prédictions correctes des relations non-nulles. Pour évaluer les performances de notre modèle, selon les cas, nous soumettons nos prédictions au service d'évaluation en ligne officiel (cas de BB-Rel) ou utilisons le kit d'évaluation officiel [7].

4. Ce modèle obtient 0,60 comme score strict et 0.78 comme score de Wang pour les habitats pour la tâche BB-Norm.
5. https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding
6. Inclure la relation nulle, qui est généralement majoritaire, donne des résultats optimistes.
7. ChemProt : https://biocreative.bioinformatics.udel.edu/tasks/biocreative-vi/track-5/ ; DrugProt : https://codalab.lisn.upsaclay.fr/competitions/8293#participate ; BB-Rel : http://bibliome.jouy.inra.fr/demo/BioNLP-OST-2019-Evaluation/index.html.

	ChemProt	DrugProt	BB-Rel$_p$
PubMedBERT	77,5 $\pm$ 0,7 / 79,6	75,1 $\pm$ 0,4 / 77,7	61,4 $\pm$ 1,2 / 64,3
KB-PubMedBERT	**78,4** $\pm$ **0,9** * / 80,4	75,5 $\pm$ 0,8 / 77,9	**63,3** $\pm$ **2,1** * / **65,7**
État de l'art	78,0 / —	— / **79,7**	— / 64,8

TABLE 3 – Scores micro F1 sur les tâches d'extraction de relation. Nous rapportons a/b où a représente le score moyen de 5 exécutions avec différentes initialisations aléatoires et b représente le score d'un ensemble à vote majoritaire. Nous rapportons les deux scores pour mieux comparer nos résultats aux résultats de l'état de l'art. * indique qu'un test T unilatéral émettant l'hypothèse que la performance moyenne de KB-PubMedBERT est meilleure que la base de comparaison obtient une p-valeur $p < 0{,}1$ (respectivement $0{,}07$ pour ChemProt et $0{,}08$ pour BB-Rel$_p$), ce qui est interprété comme une faible présomption contre l'hypothèse nulle. Le système état de l'art considéré est indiqué en début de section 4.3.

4.3 Résultats

Nous comparons KB-PubMedBERT à PubMedBERT et à l'état de l'art suivant sur chaque corpus :

1. Pour ChemProt : nous prenons comme état de l'art le modèle qui a obtenu le score le plus élevé en utilisant le même kit d'évaluation que nous. Il s'agit de SciFive (Phan *et al.*, 2021), un modèle T5 pré-entraîné sur la littérature biomédicale ;
2. Pour DrugProt : un ensemble de 10 RoBERTa-large-PM-M3-Voc (Lewis *et al.*, 2020) avec des définitions de produits chimiques sélectionnées à partir de CTD (Weber *et al.*, 2022) ;
3. Pour BB-Rel$_p$: un modèle Transformer à 12 couches pré-entraîné sur les corpus Wikipédia anglais, BooksCorpus, PubMed et PubMed Central (PMC) (Zhang *et al.*, 2019a).

Les résultats expérimentaux sont résumés dans le tableau 3. Nous observons que KB-PubMedBERT surpasse systématiquement la base de comparaison, ce qui prouve l'efficacité de notre méthode d'injection d'informations de BC. Cependant, la différence entre KB-PubMedBERT et la base de comparaison n'est pas significative sur DrugProt. Notre modèle surpasse l'état de l'art antérieur sur ChemProt et BB-Rel$_p$, mais reste environ 2 % derrière le meilleur score sur DrugProt. L'absence d'amélioration de KB-PubMedBERT sur DrugProt peut s'expliquer par le fait que près de 1 % des occurrences des entités de DrugProt sont absentes de la BC, alors que ce nombre est de $0{,}1$ % pour ChemProt et de 0 % pour BB-Rel$_p$. Comme les plongements pour ces entités absentes de la BC sont initialisés aléatoirement, un pourcentage plus élevé d'occurrences d'entités absentes signifie que moins d'informations venant de la BC sont intégrées.

4.4 Étude d'ablation complémentaire

L'examen de PubMedBERT seul ci-dessus constitue une première étude d'ablation dans laquelle le composant de plongement de graphes de notre modèle n'est pas utilisé. Inversement, pour vérifier la contribution intrinsèque du composant de plongement de graphes à la performance finale de l'extraction de relations, nous menons des expériences où nous supprimons complètement PubMedBERT de notre modèle : nous n'utilisons que la paire d'entités (source, cible) pour déduire le type d'interaction via les plongements de graphes RotatE. Les résultats sont dans le tableau 4. Nous observons que même sans aucun contexte, la base de connaissances obtient dans notre modèle un score F1 supérieur

	ChemProt	**DrugProt**	**BB-Rel$_p$**
KB-PubMedBERT$^-$	23,8 $\pm$ 1,6	19.5 $\pm$ 1,0	26,6 $\pm$ 0,3
majority	17,3	12,3	38,3

TABLE 4 – Ablation : KB-PubMedBERT$^-$ désigne notre modèle sans PubMedBERT. Nous rapportons le score moyen de 5 exécutions.

à 0,20 (la relation nulle est exclue de l'évaluation). Nous établissons par ailleurs un modèle simple qui prédit toujours la classe majoritaire. Ce modèle est nommé "*majority*" et ses résultats sont également montrés dans le tableau. On constate que les résultats de KB-PubMedBERT$^-$ sont significativement meilleurs que ceux du modèle aléatoire sur ChemProt et DrugProt, cela confirme que le profil de scores $\mathbf{h}_{score}$ représentant des suggestions de relations à grain fin de la base de connaissances entre deux entités, obtenues à partir des plongements de graphes RotatE, est utile pour l'extraction de relations biomédicales. Le fait que *majority* donne un meilleur résultat que KB-PubMedBERT$^-$ dans le cas de BB-Rel$_p$ peut être expliqué par le fait qu'il n'existe dans ce cas qu'une seule relation positive.

5 Conclusion

Dans cet article, nous proposons l'architecture KB-PubMedBERT qui injecte dans PubMedBERT les informations d'une base de connaissances pour améliorer ses performances en extraction de relations biomédicales. À la différence des modèles antérieurs utilisant des bases de connaissances, nous calculons d'abord à l'aide de la méthode de plongement de graphe RotatE les possibilités de relations de la BC entre les entités considérées, puis nous utilisons ces possibilités pour déduire les relations cibles de l'extraction de relation. Nous menons des expériences sur trois tâches d'extraction de relations biomédicales : notre modèle y surpasse systématiquement PubMedBERT et obtient des performances proches de l'état de l'art antérieur ou meilleures que lui. Une étude d'ablation confirme de plus la pertinence des relations de la base de connaissances indépendamment du modèle de langue PubMedBERT. À l'avenir, nous complèterons nos expériences sur d'autres jeux de données pour étendre l'étude de l'applicabilité de notre méthode.

Remerciements

Nous remercions la plateforme Saclay-IA de l'Université Paris-Saclay pour les ressources de calcul et de stockage du cluster GPU Lab-IA.

Ce travail a été financé par le Labex DigiCosme (projet ANR-11-LABEX-0045-DIGICOSME) opéré par l'ANR dans le cadre du programme "Investissement d'Avenir" Idex Paris-Saclay (ANR-11-IDEX-0003-02).

Références

BELTAGY I., LO K. & COHAN A. (2019). SciBERT : A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3615–3620, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1371.

BODENREIDER O. (2004). The Unified Medical Language System (UMLS) : Integrating biomedical terminology. *Nucleic Acids Research*, **32**(Database issue), D267–270. DOI : 10.1093/nar/gkh061.

BORDES A., USUNIER N., GARCIA-DURÁN A., WESTON J. & YAKHNENKO O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, p. 2787–2795, Red Hook, NY, USA : Curran Associates, Inc.

BOSSY R., DELÉGER L., CHAIX E., BA M. & NÉDELLEC C. (2019). Bacteria Biotope at BioNLP Open Shared Tasks 2019. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, p. 121–131, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-5719.

DAVIS A. P., GRONDIN C. J., JOHNSON R. J., SCIAKY D., WIEGERS J., WIEGERS T. C. & MATTINGLY C. J. (2023). Comparative Toxicogenomics Database (CTD) : update 2023. *Nucleic acids research*, **51**(D1), D1257–D1262. DOI : 10.1093/nar/gkac833.

DÉROZIER S., BOSSY R., DELÉGER L., BA M., CHAIX E., HARLÉ O., LOUX V., FALENTIN H. & NÉDELLEC C. (2023). Omnicrobe, an open-access database of microbial habitats and phenotypes using a comprehensive text mining and data fusion approach. *PloS one*, **18**(1), e0272473. DOI : 10.1371/journal.pone.0272473.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

EL BOUKKOURI H., FERRET O., LAVERGNE T., NOJI H., ZWEIGENBAUM P. & TSUJII J. (2020). CharacterBERT : Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 6903–6915, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.609.

ELSAHAR H., VOUGIOUKLIS P., REMACI A., GRAVIER C., HARE J., LAFOREST F. & SIMPERL E. (2018). T-REx : A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan : European Language Resources Association (ELRA).

FERRÉ A., DELÉGER L., BOSSY R., ZWEIGENBAUM P. & NÉDELLEC C. (2020). C-Norm : a neural approach to few-shot entity normalization. *BMC bioinformatics*, **21**(23), 579. DOI : 10.1186/s12859-020-03886-8.

GROVER A. & LESKOVEC J. (2016). node2vec : Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 855–864, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/2939672.2939754.

GU Y., TINN R., CHENG H., LUCAS M., USUYAMA N., LIU X., NAUMANN T., GAO J. & POON H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, **3**(1), 1–23. DOI : 10.1145/3458754.

HAO B., ZHU H. & PASCHALIDIS I. (2020). Enhancing clinical BERT embedding using a biomedical knowledge base. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 657–661, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.57.

IINUMA N., MIWA M. & SASAKI Y. (2022). Improving supervised drug-protein relation extraction with distantly supervised models. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, p. 161–170, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.bionlp-1.16.

KRALLINGER M., RABAL O., AKHONDI S. A., PÉREZ M. P., SANTAMARÍA J., RODRÍGUEZ G. P., TSATSARONIS G., INTXAURRONDO A., LÓPEZ J. A., NANDAL U. *et al.* (2017). Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, p. 141–146.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240. DOI : 10.1093/bioinformatics/btz682.

LEWIS P., OTT M., DU J. & STOYANOV V. (2020). Pretrained language models for biomedical and clinical tasks : Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, p. 146–157, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.clinicalnlp-1.17.

MAO J. & LIU W. (2019). Integration of deep learning and traditional machine learning for knowledge extraction from biomedical literature. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, p. 168–173, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-5724.

MICHALOPOULOS G., WANG Y., KAKA H., CHEN H. & WONG A. (2021). UmlsBERT : Clinical domain knowledge augmentation of contextual embeddings using the Unified Medical Language System Metathesaurus. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1744–1753, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.139.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In Y. BENGIO & Y. LECUN, Éds., *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

MIRANDA A., MEHRYARY F., LUOMA J., PYYSALO S., VALENCIA A. & KRALLINGER M. (2021). Overview of DrugProt BioCreative VII track : quality evaluation and large scale text mining of drug-gene/protein relations. In *Proceedings of the seventh BioCreative challenge evaluation workshop*, p. 11–21.

PHAN L. N., ANIBAL J. T., TRAN H., CHANANA S., BAHADROGLU E., PELTEKIAN A. & ALTAN-BONNET G. (2021). SciFive : a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv :2106.03598*.

RIBEIRO L. F., SAVERESE P. H. & FIGUEIREDO D. R. (2017). struc2vec : Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, p. 385–394.

SUN Z., DENG Z., NIE J. & TANG J. (2019). RotatE : Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* : OpenReview.net.

SUNG M., JEON H., LEE J. & KANG J. (2020). Biomedical entity representations with synonym marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 3641–3650, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.335.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30 de *NIPS'17*, p. 6000–6010, Red Hook, NY, USA : Curran Associates, Inc.

WANG R., TANG D., DUAN N., WEI Z., HUANG X., JI J., CAO G., JIANG D. & ZHOU M. (2021). K-Adapter : Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, p. 1405–1418, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.findings-acl.121.

WEBER L., SÄNGER M., GARDA S., BARTH F., ALT C. & LESER U. (2022). Chemical–protein relation extraction with ensembles of carefully tuned pretrained language models. *Database*, **2022**. baac098, DOI : 10.1093/database/baac098.

YUAN Z., LIU Y., TAN C., HUANG S. & HUANG F. (2021). Improving biomedical pretrained language models with knowledge. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, p. 180–190, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.bionlp-1.20.

ZHANG Q., LIU C., CHI Y., XIE X. & HUA X. (2019a). A multi-task learning framework for extracting bacteria biotope information. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, p. 105–109, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-5716.

ZHANG Z., HAN X., LIU Z., JIANG X., SUN M. & LIU Q. (2019b). ERNIE : Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 1441–1451, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1139.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Derrière les plongements de relations

Hugo Thomas[1] Guillaume Gravier[1] Pascale Sébillot[2]
(1) Univ Rennes, CNRS, Inria - IRISA, Campus de Beaulieu 35042 Rennes, France
(2) Univ Rennes, CNRS, Inria, INSA Rennes - IRISA, Campus de Beaulieu 35042 Rennes, France
hugo.thomas@irisa.fr, guillaume.gravier@irisa.fr,
pascale.sebillot@irisa.fr

RÉSUMÉ

Dans cet article, plutôt que nous arrêter aux scores de performance habituellement fournis (par ex. mesure F1), nous proposons une analyse approfondie, selon différents critères, des modélisations de relations employées par plusieurs architectures de modèles de typage de relations. Cette analyse vise à mieux comprendre l'organisation de l'espace latent des modélisations et ses propriétés, enjeu important pour les modèles se fondant sur les distances dans cet espace. Dans cet objectif d'analyse des plongements, nous étudions l'influence, sur ces modélisations, du vocabulaire, de la syntaxe, de la sémantique des relations, de la représentation des entités nommées liées, ainsi que la géométrie de leur espace latent. Il en ressort que les modélisations de relations sont apprises de manière inégale d'un modèle à un autre entraînés de la même manière ; dans ce cas, les indicateurs que nous proposons sont de nouveaux éléments de compréhension de l'espace latent d'un modèle afin de mieux exploiter ses propriétés.

ABSTRACT

What hides behind relation embeddings ?

In this paper, rather than focusing on the performance scores usually provided (e.g., the F1 measure), we propose a more in-depth analysis, according to several criteria, of the relation modeling of different model architectures for relation typing. This analysis aims to better understand the organization and properties of the latent modeling space, an important issue for models exploiting distances in this vector space. In order to study these modelings, we evaluate the influence on these models of the lexicon, the syntax, and the semantics of relations, the representation of the entities, as well as the geometry of their latent spaces. It appears that the relation modelings are learned unevenly from one model to another trained in the same way ; in this case, the indicators we proposed are additional knowledge about the latent space to better exploit its properties.

MOTS-CLÉS : Traitement automatique des langues, extraction de relations, classification de relations, modélisation de relations.

KEYWORDS: Natural language processing, relation extraction, relation classification, relation modeling.

1 Introduction

L'extraction de relations entre entités – en particulier entre entités nommées (noms de personnes, lieux, entreprises...) – est un champ de recherche très actif du traitement automatique des langues (TAL)

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 311–322.

(*cf.* par exemple (Nasar *et al.*, 2022) pour un résumé récent de nombreux travaux et des techniques utilisées). Pouvoir acquérir automatiquement et exploiter les relations décrites au sein de textes est un enjeu important, en particulier pour peupler des bases de connaissance et pour comprendre les interactions entre entités d'intérêt, et ce, dans de nombreux domaines (médecine (Karaa *et al.*, 2021), finance (Jabbari *et al.*, 2020), justice (Hong *et al.*, 2021), journalisme (Riedel *et al.*, 2010), publications scientifiques (Bhattacharya & Getoor, 2007)...).

Les relations considérées sont fréquemment binaires, exprimées au sein d'une phrase – ce sera aussi le focus de cet article – et sont modélisées par un triplet constitué d'une **entité tête** (ou sujet), d'une **entité queue** (ou objet) et d'une **relation** ou prédicat qui les lie. Par exemple, la phrase *Joe Biden est président des États-Unis depuis 2021* exprime le fait que l'entité *Joe Biden* et l'entité *États-Unis* sont liées par une relation modélisable par le triplet *(Joe Biden, est_président_de, États-Unis)*.

La littérature actuelle du TAL concernant l'extraction de relations se focalise sur deux tâches de classification principales : la première est la *détection* de relations qui consiste à déterminer si deux entités dans un texte (ou une phrase) donné sont ou non liées par une relation quelconque ; la seconde est le *typage* de relations, reposant sur la détermination du type de la relation entre deux entités supposées liées par une relation dans un texte donné.

Pour la suite de l'article, notre attention se porte sur la tâche de typage de relations : comme expliqué ensuite, nous nous focalisons sur les modélisations des relations, et nous supposerons donc que les relations dans le texte ont déjà été détectées afin de pouvoir directement les étudier en étant certains de leur présence. Cette tâche de typage se traduit alors dans le cas général par la résolution de

$$\arg\max_{r} \; p(r \mid s, e_{tête}, e_{queue}) \tag{1}$$

où s est la phrase étudiée, r est un type de relation, $\mathcal{R}$ est l'ensemble des types de relations, $(e_{tête}, e_{queue})$ sont les entités et $\mathcal{E}$ est l'ensemble des entités ($r \in \mathcal{R}$ et $(e_{tête}, e_{queue}) \in \mathcal{E}^2$).

Une étude extensive des méthodes de détection et de typage de relations (Nasar *et al.*, 2022) dégage les grandes familles de modèles et architectures adaptées à ces tâches. Elle n'aborde toutefois pas les modèles Transformers qui désormais font l'état de l'art dans un grand nombre de tâches de TAL, probablement par manque de recul sur ces modèles encore récents. La comparaison des modèles se limite à celle de leurs performances évaluées à l'aide de métriques classiques telles que la précision, le rappel et la mesure F1, qui ne suffisent toutefois pas à comprendre la nature des différences entre les modélisations. Du côté des modèles Transformers, des avancées notables sont faites en termes de scores de classification, mais encore une fois, sans qu'on ne dispose d'une analyse fine de la modélisation des relations sous-jacente. C'est par exemple le cas de modèles tels que LUKE (Yamada *et al.*, 2020) ou le modèle proposé par (Zhou & Chen, 2022), qui figurent parmi l'état de l'art du typage de relations et proposent une modélisation novatrice des entités sans approfondir la modélisation résultante des relations. Enfin, les modèles récents d'extraction de relations non supervisée ou sans exemple (*zero shot*) s'appuient sur des encodeurs Transformers et notamment sur la mesure de similarité dans leur espace latent (par exemple (Chen & Li, 2021) ou plusieurs modèles de l'état de l'art décrit par Simon (2022)). Il est dans ce cas primordial de comprendre les attributs et informations de la relation reflétés par leur représentation, afin de saisir sur quoi s'appuie un modèle pour définir la similarité entre deux relations. Sans cela, il est impossible de comprendre les erreurs du modèle, ou au contraire ce sur quoi repose son efficacité.

Il apparaît donc qu'une meilleure appréhension des modélisations des relations apprises par les modèles de typage de relations manque dans ce domaine du TAL. S'arrêter à des scores de classification

est une manière objective et efficace de comparer la performance de modèles, mais, dans une optique académique, est insuffisant pour comprendre la nature des différences de représentations entre ces classifieurs. Il est donc nécessaire d'enrichir la comparaison des modèles avec des indicateurs plus fins. Par exemple, bien que des modèles aient la capacité de tirer parti de la syntaxe des relations, la littérature récente ne vérifie pas l'influence de celle-ci sur la modélisation résultante. L'impact de la représentation des entités liées n'est également pas éprouvé au-delà des scores de classification. L'espace latent des modélisations de relations de chaque modèle est le siège de l'information sur ces relations, mais il n'est pas sondé au-delà des visualisations de ses projections en deux dimensions par t-SNE.

Pour améliorer les comparaisons des modèles, l'initiative de Alt *et al.* (2020) d'étude critique du jeu de données TACRED procède à une analyse approfondie des erreurs de quatre modèles (taux d'erreurs sur les longues relations, les relations avec le même type d'entités tête et queue, les relations aux entités éloignées...). Nous nous positionnons dans la lignée de cette démarche, mais souhaitons aller plus loin et comparer des modèles représentatifs d'architectures plus hétérogènes sur davantage de critères afin d'aboutir à une analyse plus fine.

Dans un premier temps, nous présentons les jeux de données utilisés et les modèles comparés, afin d'exposer leurs différences pour contextualiser les résultats de nos expériences. Nous examinons dans un deuxième temps plusieurs critères : l'influence de la prise en compte de la syntaxe sur le typage des relations ; la représentation des entités et la façon dont elle affecte les représentations de relations ; l'intensité du lien entre la similarité des plongements de deux phrases supports de relations et la similarité lexicale, syntaxique et sémantique de ces deux phrases, ainsi que les distances et dispersions inter-relations dans l'espace latent de représentation des relations. Nous explorons ces critères pour déceler des différences de plusieurs natures entre les modélisations étudiées sur les jeux de données choisis. Nous concluons avec un résumé des résultats des expériences et leurs implications grâce au recul apporté.

2 Contexte et cadre expérimental

Notre analyse porte sur une variété de modèles de typage et extraction de relations représentatifs de différentes avancées du domaine. L'objectif étant de proposer des éléments de comparaison fiables et fins entre ces modèles, nous choisissons d'homogénéiser l'apprentissage de ceux-ci. Ainsi, tous les classifieurs ont pour point commun qu'une relation est modélisée par un vecteur de dimension fixe $\mathcal{D}$. Le deuxième point commun est l'architecture de la tête de classification utilisée pour déduire de chacune de ces modélisations une prédiction de type de relation par l'équation 1. Cette tête de classification effectue une prédiction sous la forme du vecteur $x_{pred} \in \mathbb{R}^n$ (n est le nombre de types de relation égal à $card(\mathcal{R})$) pour la phrase support d'une relation modélisée par x_{rel}, tel que $x_{pred} = W_2 ReLU(W_1 x_{rel} + b_1) + b_2$. W_1 et W_2 sont les matrices des poids des deux couches de neurones, et b_1 et b_2 les biais de ces mêmes neurones. $x_{rel} \in \mathbb{R}^{\mathcal{D}}$, $W_1 \in \mathbb{R}^{\mathcal{D}\times\mathcal{D}}$, $W_2 \in \mathbb{R}^{\mathcal{D}\times n}$, $b_1 \in \mathbb{R}^{\mathcal{D}}$, $b_2 \in \mathbb{R}^n$, et nous choisissons $\mathcal{D} = 768$.

L'hypothèse de domaine clos est admise, c'est-à-dire que la tâche de typage de relations est restreinte à identifier une relation parmi $card(\mathcal{R})$ types et aucun autre type de relation n'est supposé possible. Ce contexte permet de se focaliser sur des relations précises et nommées afin de faciliter l'étude de celles-ci.

Une première modélisation que nous examinons s'appuie sur la similarité lexicale (de l'expression) des relations, des (portions de) phrases supports d'une même relation utilisant supposément un vocabulaire voisin. Ce modèle n'est pas neuronal, mais on applique sur sa modélisation des couches de neurones de prédiction du type de la relation, comme décrit au premier paragraphe. Une phrase est donc traitée comme un vecteur sac-de-mots ayant une composante par mot du dictionnaire, pondéré par TF-IDF (*term frequency-inverse document frequency*). Nous concaténons à la fin de ce vecteur l'encodage *one hot* du type des deux entités de la relation. Ce vecteur concaténé est ramené par analyse en composantes principales à la dimension désirée $\mathcal{D}$ pour obtenir la modélisation finale. On remarquera que la notion de séquence est perdue dans ce modèle, ce qui rend, pour lui, impossible l'exploitation de la syntaxe des phrases.

Un deuxième modèle, popularisé par le domaine de la vision par ordinateur, est le réseau de neurones convolutif (Kim, 2014) : différentes tailles de noyaux de convolution sont appliquées sur les plongements des *tokens* en entrée pour extraire de l'information à plusieurs échelles. La valeur maximale en sortie de chaque strate de traitement est extraite pour obtenir un vecteur par taille de noyau de convolution. Ces vecteurs sont ensuite concaténés, puis leur dimension est réduite par des couches de neurones artificiels. En sortie, la modélisation des relations pour ce modèle est obtenue. Cette méthode repose en grande partie sur des schémas répétés dans les phrases d'entraînement et peut donc souffrir en cas de grande variabilité des formes de surface exprimant un même type de relation.

La troisième modélisation concerne les modèles de réseaux de neurones récurrents, qui traitent de façon séquentielle l'information. On leur reproche de noyer l'information en début de séquence, ainsi que de souffrir de l'évanescence du gradient, mais ce sont des problèmes que les LSTM bidirectionnels ou Bi-LSTM atténuent. Ainsi, nous considérons dans notre étude la concaténation des vecteurs cachés des deux LSTM (avant et arrière) comme modélisation de relation.

Les Transformers, encodeurs-décodeurs tirant parti des mécanismes d'attention, offrent depuis 2017 (Vaswani *et al.*, 2017) des performances élevées sur les données textuelles dans des tâches requérant une compréhension fine du langage, par exemple la traduction ou l'analyse des sentiments. On étudie l'état du vecteur associé au token [CLS] en sortie du modèle comme représentation du texte en entrée. Nous examinons les modèles RoBERTa (Liu *et al.*, 2019), ainsi que LUKE (Yamada *et al.*, 2020) qui possède des plongements et des mécanismes d'attention spécifiques aux entités pour améliorer la classification de relations. Pour nos entraînements, les Transformers ne sont ni appris à partir de poids aléatoires, ni ajustés entièrement à partir de modèles pré-entraînés : nous adaptons les modèles en entraînant des couches de réduction et augmentation de dimensionnalité dans chaque bloc Transformer comme décrit dans (Houlsby *et al.*, 2019). Cette méthode permet l'apprentissage d'une fraction des paramètres originaux du modèle pour des performances similaires. Le modèle RoBERTa base est adapté pour typer les relations, et le modèle RoBERTa+entités est adapté pour typer les relations et les entités tête et queue. Nous considérons LUKE uniquement comme référence pour ses scores, mais l'analyse de sa modélisation n'est pas effectuée, n'étant pas compatible à l'adaptation et nos tentatives de *fine-tuning* ayant échoué.

Enfin, nous étudions la modélisation des relations d'un Transformer appris de manière supervisée pour une tâche proxy, c'est-à-dire la prédiction du type des entités de la relation (entreprise, personne, lieu...), dont les poids sont ensuite gelés pour apprendre la tête de typage de relation : ce modèle est analogue à celui de Huang & Wong (2020), avec pour encodeur un Transformer et non des LSTM. Puisque les types d'entités déterminent en partie le type de la relation, il est réaliste d'estimer que cette tâche proxy de prédiction d'entités apprend des attributs pertinents pour la classification de relations.

Les fonctions utilisées pour entraîner ces différents modèles sont disponibles sur le dépôt de code de l'article [1].

Les réseaux de neurones sont entraînés par descente de gradient optimisée par Adam avec un taux d'apprentissage de 0.001 par *batch* de 8 relations pendant 128 époques avec interruption précoce (conditionnée sur la mesure F1 du modèle sur le jeu de validation). Chaque modèle est entraîné cinq fois avec une graine aléatoire différente. Les plongements de mots de chaque modèle (sauf TF-IDF+ACP qui n'en nécessite pas) sont ceux du modèle préentraîné RoBERTa base.

Afin d'effectuer selon plusieurs critères l'analyse des modélisations sous-jacentes de modèles, nous employons deux jeux de données conçus pour la tâche de typage de relation, dont les caractéristiques sont déclinées dans le tableau 1.

TACRED (Zhang *et al.*, 2017) (*TAC Relation Extraction Dataset*) présente 41 types de relations dans 106 264 exemples en anglais. 79,5% de ces relations sont du type « no_relation » (pas de relation entre les deux entités courantes dans une phrase), et les fréquences des types de relations sont peu équilibrées, présentant ainsi des classes très majoritaires et minoritaires (cadre de classification frugale). Nous ignorons la classe « no_relation » car nous étudions la tâche de typage de relations et pas de détection de relations.

FewRel (Han *et al.*, 2018), également en langue anglaise, contient 100 types de relations dans 56 000 exemples, et sa spécificité est son évaluation : il est conçu pour être évalué sur la classification frugale, avec peu d'exemples. Le script d'évaluation échantillonne le jeu de validation selon le schéma *N-way K-shot*, c'est-à-dire la classification parmi N classes avec seulement K exemples par classe. Souhaitant nous focaliser sur la modélisation des relations des modèles à leur plein potentiel, l'aspect de classification frugale n'est pas pris en compte dans cet article ; ainsi, nous découpons le jeu d'entraînement en deux parties (jeu d'entraînement raccourci et jeu de test) afin d'obtenir un jeu de test dédié en plus des jeux d'entraînement et de validation.

jeu de données	jeu d'entraînement	jeu de validation	jeu de test	total
TACRED	13 012	5436	3325	21 773
FewRel 1.0	33 600	11 200	11 200	56 000

TABLE 1 – Résumé du contenu des jeux de données.

3 Expériences

Nous procédons par étape aux différentes analyses des modélisations de relations en commençant par étudier l'impact de la syntaxe et de sa prise en compte sur les modélisations obtenues (et donc sur la qualité des représentations pour la tâche de typage de relations). Nous nous penchons ensuite sur l'influence de la représentation choisie des entités de la relation sur ces modélisations. Enfin, pour aller au-delà d'une analyse s'étayant à l'aune de la seule qualité plus ou moins forte en classification, nous proposons une étude comparative des modélisations de relations obtenues d'un point de vue langue (deux relations proches en termes de représentations vectorielles ont-elles des « propriétés langagières » proches ?) et d'un point de vue géométrie des espaces vectoriels appris.

1. Dépôt disponible sur ce lien : https://gitlab.inria.fr/huthomas/taln_experiments

3.1 Impact de la syntaxe

L'identification des relations peut reposer en partie sur la syntaxe des phrases (ou parties de phrases) supports de celles-ci. Étant donné que le typage de la relation dépend de la modélisation, nous souhaitons vérifier l'importance de la syntaxe sur cette modélisation de relation ; nous procédons à une comparaison des différents modèles en conservant d'une part la phrase entière en entrée, d'autre part uniquement les mots du plus court chemin de dépendance syntaxique entre les deux entités de la relation, dans l'ordre de ce chemin, de l'entité tête à l'entité queue.

modèle	partie de phrase considérée	F1 sur TACRED		F1 sur FewRel	
		moyenne micro	moyenne macro	moyenne micro	moyenne macro
TF-IDF+ACP	plus court chemin	**83.3±0.62%**	**61.19±0.75%**	62.32±0.51%	61.92±0.61%
	phrase entière	**84.14±0.64%**	**60.83±1.31%**	61.79±0.59%	61.16±0.63%
CNN	plus court chemin	74.07±0.73%	53.85±0.91%	**71.04±0.22%**	**70.98±0.23%**
	phrase entière	42.1±0.93%	27.39±0.64%	54.24±0.41%	52.91±0.67%
Bi-LSTM	plus court chemin	72.67±1.13%	53.71±1.79%	67.42±2.05%	66.64±2.41%
	phrase entière	37.91±0.68%	23.48±1.32%	49.74±1.07%	47.34±1.48%
RoBERTa base	plus court chemin	77.89±0.37%	58.96±1.53%	**71.43±0.37%**	**70.55±0.9%**
	phrase entière	43.17±0.54%	28.66±0.91%	58.43±0.95%	56.06±1.15%
RoBERTa+entités	plus court chemin	76.05±1.13%	56.36±0.98%	**73.75±0.81%**	**73.14±0.91%**
	phrase entière	42.17±1.09%	29.06±0.77%	56.14±0.54%	54±0.63%
RoBERTa proxy	plus court chemin	72.49±1.64%	50.73±2.2%	55±0.96%	53.54±1.15%
	phrase entière	37.38±1.69%	23.63±1.28%	48.97±0.47%	47.16±0.48%

TABLE 2 – Comparaison des résultats de typage de relations en considérant la phrase entière ou seulement le plus court chemin de dépendance syntaxique entre les deux entités de la phrase. Les meilleurs scores au risque de 5% sont indiqués en gras.

Les résultats sont présentés dans le tableau 2. En étudiant d'abord le modèle TF-IDF+ACP sur les premières lignes, nous observons une faible différence entre les deux configurations sur les deux jeux de données : ce modèle ne prend pas en compte la syntaxe, et cette différence est seulement due au nombre de mots inférieur dans le cas du plus court chemin de dépendance. Les scores de ce modèle sont les meilleurs sur TACRED – signifiant la forte dépendance des types de relations au vocabulaire dans ce jeu de données – et sont bons sur FewRel. Le vocabulaire influe donc en partie sur le typage de relations, mais la syntaxe est nécessaire pour de meilleurs scores, notamment sur FewRel. En se penchant sur les autres lignes du tableau, nous constatons que les autres modèles prenant la syntaxe en compte sont systématiquement meilleurs lorsque les entrées sont les plus courts chemins de dépendance syntaxique : cette normalisation grammaticale des relations est bénéfique, démontrant l'importance de la prise en compte de la syntaxe pour la modélisation des relations dans l'objectif de leur typage.

3.2 Influence des représentations des entités

Nous examinons ici l'impact de la représentation des entités sur la modélisation résultante des relations. En effet, les entités sont les sujets et objets mêmes des relations étudiées, et y jouent donc très vraisemblablement un rôle-clé. Les entités sont, dans un premier temps, conservées telles qu'elles apparaissent dans la forme de surface où s'exprime la relation. Ensuite, elles sont entourées de balises afin de signifier au modèle où elles commencent et finissent dans la phrase. Enfin, les entités sont remplacées par leur type (lieu, personne, entreprise...) pour vérifier s'il n'est pas suffisant pour prédire la relation. Le modèle TF-IDF+ACP est exclu, car ajouter les balises d'entités dans son vocabulaire

ne changera rien puisqu'elles apparaissent identiquement dans chaque phrase ; de plus, le modèle bénéficie déjà du type des entités parmi ses attributs.

modèle	représentation des entités	F1 sur TACRED		F1 sur FewRel	
		moyenne micro	moyenne macro	moyenne micro	moyenne macro
CNN	entité non modifiée	42.1±0.93%	27.39±0.64%	54.24±0.41%	52.91±0.67%
	type de l'entité	79.19±0.34%	61.78±0.92%	56.67±0.35%	55.9±0.32%
	entité balisée	69.71±0.7%	52.67±1.13%	66.37±0.35%	65.55±0.34%
Bi-LSTM	entité non modifiée	37.91±0.68%	23.48±1.32%	49.74±1.07%	47.34±1.48%
	type de l'entité	**80.87±0.35%**	**64.39±0.7%**	53.7±1.81%	52.52±1.61%
	entité balisée	69.95±0.94%	54.48±1.29%	63.34±0.35%	62.17±0.62%
RoBERTa base	entité non modifiée	43.17±0.54%	28.66±0.91%	58.43±0.95%	56.06±1.15%
	type de l'entité	**80.86±0.27%**	**65.62±1.01%**	63.35±0.5%	61.89±0.78%
	entité balisée	77.43±0.91%	61.15±1.34%	**73.79±1.24%**	**72.65±1.47%**
RoBERTa+entités	entité non modifiée	42.17±1.09%	29.06±0.77%	56.14±0.54%	54±0.63%
	type de l'entité	**79.75±0.36%**	**64.39±1.79%**	64.47±0.78%	63.04±1.12%
	entité balisée	74.45±0.59%	57.89±2.88%	**75.16±0.13%**	**74.09±0.12%**
RoBERTa proxy	entité non modifiée	33.12±8.29%	19.47±8.52%	48.97±0.47%	47.16±0.48%
	type de l'entité	71.96±1.87%	46.28±1.71%	41.67±1.51%	40.03±1.64%
	entité balisée	69.08±0.89%	51.05±2.25%	59.28±0.88%	58.03±0.61%

TABLE 3 – Comparaison des résultats en fonction de la représentation des entités dans la phrase entière. Les meilleurs scores au risque de 5% sont indiqués en gras.

Le tableau 3 résume les scores obtenus. Sur la colonne de TACRED, les lignes liées aux entités non modifiées conduisent systématiquement à des scores plus faibles que les autres configurations : les modèles n'ont pas de repères pour la position des entités et peuvent confondre celles-ci avec d'autres entités de la phrase. Baliser les entités améliore les scores en levant cette confusion. Les meilleurs scores sont obtenus en remplaçant les entités par leur type, qui apporte une information riche et condensée sur elles, au prix de la perte de leurs mentions exactes dans la phrase.

Sur FewRel, le constat reste le même pour les entités non modifiées menant à des scores faibles. Remplacer les entités par leur type améliore encore une fois les scores sauf pour RoBERTa proxy, qui focalise vraisemblablement trop son apprentissage sur le typage des entités et dégrade ses attributs pour le typage de relations. Les balises autour des entités conduisent aux meilleurs scores sur ce jeu de données, conservant la mention d'entité intacte et indiquant sa position.

D'autres représentations des entités existent et peuvent améliorer encore les scores ; par exemple, LUKE obtient sur TACRED une mesure F1 micro de 88.91% et macro de 59.82%, mais n'entre pas dans notre comparaison à cause de la différence de largeur du modèle LUKE préentrainé sur TACRED (dimension supérieure des plongements). Ces scores vont tout de même dans le sens de l'existence d'un impact des représentations des entités sur le typage de relations. Il découle de ce constat et des expériences que les entités des relations et leur représentation ont une influence importante pour le typage de celles-ci, impactant directement le score des modèles prenant en compte ces entités.

3.3 Analyse des plongements des relations

Les deux expériences précédentes se penchant uniquement sur les scores de classification (mesure F1) pour tirer des conclusions, nous complétons notre analyse avec des métriques décrivant plus finement les différences entre modélisations de relations. Nous choisissons de nous attacher tout d'abord à des mesures directement liées au langage, avant d'effectuer des mesures géométriques.

Dans un premier temps, nous voulons étudier, pour chaque modèle, à quel point la similarité entre deux plongements de relations est proche de la similarité lexicale, syntaxique ou sémantique des formes de surface exprimant ces relations. Ceci permet de nous renseigner sur le fait qu'un modèle s'appuie plus ou moins sur ces trois types d'information. Pour ce faire, nous calculons la corrélation de Spearman entre la similarité cosinus des deux plongements de relations et chacune des trois autres similarités. La similarité lexicale est calculée par un Jaccard (taille de l'intersection des mots divisée par la taille de l'union des mots) entre les (portions de) phrases exprimant les relations. La similarité syntaxique est estimée en calculant la distance entre les arbres syntaxiques des deux phrases selon le noyau décrit dans (Culotta & Sorensen, 2004). La similarité sémantique est obtenue par similarité cosinus entre les moyennes des vecteurs de sortie d'un modèle SentenceBERT (Reimers & Gurevych, 2019) (moyenne des vecteurs de sortie de l'encodeur de chaque token) de détection de paraphrases. Nous échantillonnons 3 000 paires aléatoires de phrases toutes relations confondues dans chacun des jeux de données, mesurons chaque type de similarité sur ces paires, puis calculons la corrélation de Spearman entre la similarité des plongements de relations et chacune des trois autres similarités. Pour les corrélations données au tableau 4, la forme de surface considérée pour tous les modèles est la phrase entière sans modification des entités afin de garantir des similarités fiables ; les mêmes calculs ont été effectués avec les meilleures configurations par modèle, mais donnant des résultats comparables, ne sont pas répertoriés ici par manque de place.

Dans un second temps, nous souhaitons examiner l'organisation géométrique des espaces latents de représentations des relations des différents modèles. En effet, les différents encodeurs étudiés plongent des phrases support de relation dans un espace vectoriel de dimension élevée (768), il est donc important de s'intéresser à l'organisation spatiale de ces vecteurs de relations au sein de l'espace vectoriel pour différencier ces encodeurs. Après avoir constitué des *clusters* de plongements de relations par type, nous calculons les distances entre leurs centroïdes (vecteur moyen de tous les plongements des phrases exprimant une même relation) , afin d'observer la densité relative de chaque espace vectoriel : à cette fin, nous calculons la distance moyenne de chaque centroïde à ses cinq plus proches centroïdes voisins, et établissons la moyenne de ces distances. Nous cherchons, d'autre part, à mesurer le recouvrement des *clusters*, qui traduit la confusion d'un encodeur entre plusieurs types de relations. Pour cela, pour chaque plongement de relation, ses cinq plus proches voisins sont considérés, et la moyenne, pour tous les plongements, de leurs plus proches voisins (parmi les cinq) n'étant pas du même type de relation qu'eux est calculée. À nouveau, dans le tableau 5 recensant les distances et les recouvrements, les plongements manipulés concernent les phrases entières exprimant les relations (même remarque que précédemment pour ce qui est des expériences faites avec les meilleures configurations des modèles).

Malgré les valeurs assez faibles du tableau 4, explicables par la métrique (corrélation de Spearman) qui détecte les tendances monotones mais décroît rapidement avec la dispersion des valeurs, les différences marquées entre les valeurs de corrélation permettent cependant de tirer des conclusions. Sur la colonne de TACRED, la corrélation avec la similarité sémantique est assez élevée, surtout pour les modèles CNN, RoBERTa base et RoBERTa+entités : leur modélisation s'appuie en partie sur la sémantique des relations pour les représenter. Ces trois modèles ont également des corrélations marquées avec la similarité syntaxique, alors que TF-IDF+ACP a une corrélation extrêmement faible due à son incapacité de prise en compte de la syntaxe. Quant à la similarité lexicale, les modèles RoBERTa base et RoBERTA+entités se démarquent avec des corrélations plus fortes : ces deux modèles prennent le mieux en compte les trois aspects langagiers étudiés. La colonne correspondant à FewRel donne des résultats différents. La similarité des plongements du modèle CNN est la plus fortement corrélée à la similarité sémantique, suivi des autres modèles. Seul TF-IDF+ACP

Modèles	TACRED corrélation (*100) avec similarité			FewRel corrélation (*100) avec similarité		
	lexicale	syntaxique	sémantique	lexicale	syntaxique	sémantique
TF-IDF+ACP	12.58	2.18	21.15	6.76	4.84	7.02
CNN	14.23	25.67	29.13	14.87	19.64	30.68
Bi-LSTM	16.07	11.75	22.37	6.26	10.21	23.49
RoBERTa base	25.76	34.01	31.39	7.40	14.26	18.95
RoBERTa+entités	20.86	31.42	33.43	8.43	11.42	21.30
RoBERTa proxy	16.78	15.61	24.70	5.52	12.81	18.75

TABLE 4 – Corrélation de la similarité cosinus des modélisations de relations avec différentes natures de similarité (lexicale, syntaxique et sémantique). Les corrélations sont multipliées par 100 pour améliorer la lisibilité.

modèles	TACRED		FewRel	
	distance inter-classes des centroïdes de relations	recouvrement des *clusters* de types de relations	distance inter-classes des centroïdes de relations	recouvrement des *clusters* de types de relations
TF-IDF+ACP	0.6916±0.5255	17.84%	0.5447±0.3361	64.12%
CNN	6.9164±3.8321	59.54%	7.6531±4.6236	60.97%
Bi-LSTM	0.5577±0.3038	62.92%	0.8039±0.4639	54.94%
RoBERTa base	0.7980±0.4573	57.71%	1.3204±0.7979	50.26%
RoBERTa+entités	1.1839±0.6462	56.00%	0.7464±0.4305	50.17%
RoBERTa proxy	0.5315±0.3745	65.39%	0.6276±0.3581	63.03%

TABLE 5 – Comparaison de la géométrie des espaces latents des différents modèles.

possède une très faible valeur de corrélation, qui était pourtant élevée sur TACRED ; cette chute peut être due à la complexité relative de FewRel, possédant notamment 100 types de relations contre 41 dans TACRED. La corrélation syntaxique pour le modèle CNN est la plus élevée, explicable par le fonctionnement de celui-ci, s'attachant aux motifs dans la phrase et donc possiblement à la construction grammaticale. Cette corrélation reste faible pour TF-IDF+ACP pour la même raison que sur TACRED. La corrélation avec la similarité lexicale est faible sur ce jeu de données, mais le modèle CNN se démarque encore une fois.

De manière générale sur les deux jeux de données (scores les plus élevés, restant consistants sur TACRED et FewRel), les meilleurs modèles dans leur meilleure configuration selon les tableaux 2 et 3 – soit RoBERTa base et RoBERTa+entités, et CNN dans une moindre mesure – sont parmi les plus corrélés avec les similarités sémantique, syntaxique et lexicale dans ce tableau. Ce résultat suggère des capacités langagières élevées de ces modèles, en particulier les Transformers, et leur faculté à exploiter ces informations, justifiant en partie leurs bonnes performances. Ces corrélations restent toutefois faibles dans l'absolu, n'expliquant donc pas tout, ce qui signifie que les modèles ont probablement appris une autre forme de similarité plus pertinente pour la tâche, que l'on peut nommer similarité de typage de relation.

Ce constat est renforcé par les résultats du tableau 5, révélant des recouvrements des *clusters* relativement plus faibles pour ces modèles aux scores de typage de relations élevés et aux corrélations précédentes les plus fortes. Sur TACRED, le modèle CNN possède la plus grande distance moyenne inter-classes, suivi de loin par RoBERTa+entités puis RoBERTa base, et ces modèles ont un recouvrement des *clusters* conséquent malgré leur éloignement suggéré par la métrique précédente. Les modèles Bi-LSTM et RoBERTa proxy souffrent du plus fort recouvrement de leurs *clusters* et de distances inter-classes faibles, les rendant mal adaptés à l'exploitation des distances dans leur espace

latent. À l'opposé, les types de relations modélisés par TF-IDF+ACP sont bien séparés comme le suggèrent le très faible recouvrement de leurs *clusters*. Dans la colonne de FewRel, nous constatons que les *clusters* modélisés par TF-IDF+ACP et RoBERTa proxy se recouvrent fortement, ces modèles étant donc de mauvais candidats à la recherche de relations similaires dans leur espace latent comme expliqué précédemment. Les modèles RoBERTa avec et sans typage d'entités sont au contraire de bons candidats avec les plus faibles recouvrements.

4 Conclusion

Grâce aux expériences effectuées, nous observons que la syntaxe des relations et la représentation des entités ont une influence importante sur le typage des relations : avoir un modèle capable de prendre en compte la syntaxe et adopter une représentation des entités pertinente pour la tâche visée sont donc indispensables pour obtenir la modélisation de relations la plus propice à des meilleurs scores de classification.

La comparaison des espaces latents des différentes architectures de modèles dévoile des modélisations de relations sous-jacentes inégales à plusieurs égards : la corrélation de la similarité des plongements de relations dans ces modélisations avec la similarité lexicale, syntaxique ou sémantique varie significativement d'un modèle à l'autre, révélant leurs affinités respectives avec chacun de ces trois aspects langagiers. La répartition des *clusters* de types de relations nous renseigne également sur ces espaces latents : nous constatons pour certains modèles un recouvrement important des *clusters* et des distances inter-*clusters* faibles. Dans le cas où les distances dans l'espace latent sont à exploiter – par exemple pour la classification frugale à l'aide d'un modèle préentraîné –, il est utile de prendre en compte ces affinités lexicale, syntaxique et sémantique, ainsi que les informations sur la distribution des *clusters* afin d'obtenir les natures de similarité voulues. Il peut, par exemple, être souhaitable d'éviter une similarité des plongements de relations corrélée à la similarité lexicale lorsque les phrases considérées ont un vocabulaire très semblable mais de fines différences syntaxiques ou sémantiques déterminant la classe à prédire ; il peut également être souhaitable de chercher un espace avec des *clusters* distincts lorsque l'objectif est d'utiliser la proximité des voisins d'une relation dans l'espace latent comme notion de similarité du type de relation, afin d'éviter de mauvaises prédictions causées par le recouvrement de plusieurs *clusters*.

Notre étude sur deux jeux de données met donc en lumière l'influence de la syntaxe et de la représentation des entités sur les résultats de la tâche de typage de relations, ainsi que la variété des espaces latents appris par les différentes architectures de modèles pour cette tâche, aux propriétés langagières et spatiales diverses. Elle donne des indicateurs pour la compréhension et l'exploitation d'une modélisation de relations appropriée au traitement désiré, puisque que ceux-ci sont révélateurs de ces hétérogénéités dans les représentations des relations.

Références

ALT C., GABRYSZAK A. & HENNIG L. (2020). TACRED revisited : A thorough evaluation of the TACRED relation extraction task. In *58th Annual Meeting of the Association for Computational Linguistics*, p. 1558–1569, Online.

BHATTACHARYA I. & GETOOR L. (2007). Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, **1**(1).

CHEN C.-Y. & LI C.-T. (2021). ZS-BERT : Towards zero-shot relation extraction with attribute representation learning. In *2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 3470–3479, Online.

CULOTTA A. & SORENSEN J. (2004). Dependency tree kernels for relation extraction. In *42nd Annual Meeting of the Association for Computational Linguistics*, p. 423–429, Barcelona, Spain.

HAN X., ZHU H., YU P., WANG Z., YAO Y., LIU Z. & SUN M. (2018). FewRel : A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *2018 Conference on Empirical Methods in Natural Language Processing*, p. 4803–4809, Brussels, Belgium.

HONG J., VOSS C. & MANNING C. (2021). Challenges for information extraction from dialogue in criminal law. In *1st Workshop on NLP for Positive Impact*, p. 71–81, Online.

HOULSBY N., GIURGIU A., JASTRZEBSKI S., MORRONE B., DE LAROUSSILHE Q., GESMUNDO A., ATTARIYAN M. & GELLY S. (2019). Parameter-efficient transfer learning for NLP. In *36th International Conference on Machine Learning*, p. 2790–2799, Long Beach, California, USA.

HUANG H. & WONG R. (2020). Deep embedding for relation extraction on insufficient labelled data. In *2020 International Joint Conference on Neural Networks*, p. 1–8, Glasgow, United Kingdom.

JABBARI A., SAUVAGE O., ZEINE H. & CHERGUI H. (2020). A French corpus and annotation schema for named entity recognition and relation extraction of financial news. In *12th Language Resources and Evaluation Conference*, p. 2293–2299, Marseille, France.

KARAA W. B. A., ALKHAMMASH E. H. & AIDA B. (2021). Drug disease relation extraction from biomedical literature using NLP and machine learning. *Mobile Information Systems, special issue Intelligent Data Analytics for Internet of Things-Based Applications*, **2021**.

KIM Y. (2014). Convolutional neural networks for sentence classification. In *2014 Conference on Empirical Methods in Natural Language Processing*, p. 1746–1751, Doha, Qatar.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). RoBERTa : A robustly optimized BERT pretraining approach. In *20th China National Conference on Computational Linguistics*, p. 1218–1227, Hohhot, China.

NASAR Z., JAFFRY S. W. & MALIK M. K. (2022). Named entity recognition and relation extraction : State-of-the-art. *ACM Computing Surveys*, **54**(1).

REIMERS N. & GUREVYCH I. (2019). Sentence-BERT : Sentence embeddings using siamese BERT-networks. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, p. 3982–3992, Hong Kong, China.

RIEDEL S., YAO L. & MCCALLUM A. (2010). Modeling relations and their mentions without labeled text. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, p. 148–163. Barcelona, Spain : Lecture Notes in Computer Science, vol. 6323, Springer.

SIMON E. (2022). *Deep Learning for Unsupervised Relation Extraction*. Thèse de doctorat, Sorbonne Université.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. In *31st Conference on Neural Information Processing Systems*, p. 6000–6010, Long Beach, California, USA.

YAMADA I., ASAI A., SHINDO H., TAKEDA H. & MATSUMOTO Y. (2020). LUKE : Deep contextualized entity representations with entity-aware self-attention. In *2020 Conference on Empirical Methods in Natural Language Processing*, p. 6442–6454, Online.

ZHANG Y., ZHONG V., CHEN D., ANGELI G. & MANNING C. D. (2017). Position-aware attention and supervised data improve slot filling. In *2017 Conference on Empirical Methods in Natural Language Processing*, p. 35–45, Copenhagen, Denmark.

ZHOU W. & CHEN M. (2022). An improved baseline for sentence-level relation extraction. In *2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and 12th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, p. 161–168, Online.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

CamemBERT-bio : Un modèle de langue français savoureux et meilleur pour la santé

Rian Touchent Laurent Romary Éric Villemonte de la Clergerie
INRIA, 2 rue Simone IFF, 75012, France
{prénom.nom}@inria.fr

RÉSUMÉ

Les données cliniques dans les hôpitaux sont de plus en plus accessibles pour la recherche à travers les entrepôts de données de santé, cependant ces documents sont non-structurés. Il est donc nécessaire d'extraire les informations des comptes-rendus médicaux. L'utilisation du transfert d'apprentissage grâce à des modèles de type BERT comme CamemBERT ont permis des avancées majeures, notamment pour la reconnaissance d'entités nommées. Cependant, ces modèles sont entraînés pour le langage courant et sont moins performants sur des données biomédicales. C'est pourquoi nous proposons un nouveau jeu de données biomédical public français sur lequel nous avons poursuivi le pré-entraînement de CamemBERT. Ainsi, nous présentons une première version de CamemBERT-bio, un modèle public spécialisé pour le domaine biomédical français qui montre un gain de 2,54 points de F-mesure en moyenne sur différents jeux d'évaluations de reconnaissance d'entités nommées biomédicales.

ABSTRACT

a Tasty French Language Model Better for your Health

Clinical data in hospitals are increasingly accessible for research through clinical data warehouses, however these documents are unstructured. It is therefore necessary to extract information from medical reports to conduct clinical studies. Transfer learning with BERT-like models such as CamemBERT has allowed major advances, especially for named entity recognition. However, these models are trained for plain language and are less efficient on biomedical data. This is why we propose a new French public biomedical dataset on which we have continued the pre-training of CamemBERT. Thus, we introduce a first version of CamemBERT-bio, a specialized public model for the French biomedical domain that shows 2.54 points of F1 score improvement on average on different biomedical named entity recognition tasks.

MOTS-CLÉS : comptes-rendus médicaux, TAL clinique, CamemBERT, extraction d'information, biomédical, reconnaissance d'entités nommées.

KEYWORDS: EHR, clinical NLP, CamemBERT, information extraction, biomedical, named entity recognition.

1 Introduction

On observe ces dernières années un développement des entrepôts de données de santé (EDS) dans les hôpitaux. Ce sont des bases de données cliniques ayant pour but d'être plus accessibles pour la recherche. Ces documents représentent une opportunité pour des études cliniques massives sur des

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 323–334.

données réelles. Elles peuvent prendre plusieurs formes, comme des comptes-rendus, des imageries médicales ou encore des prescriptions. Cependant c'est dans les comptes-rendus que la plupart des informations se trouvent. On estime que jusqu'à 80% des entités sont absentes des autres modalités (Raghavan *et al.*, 2014). Ces données, bien que très riches, sont non-structurées, ce qui implique un pré-traitement avant de pouvoir être utilisées dans une étude clinique.

Modèles de langue pour l'extraction d'information Les modèles de type BERT (Devlin *et al.*, 2019) montrent de manière consistante des résultats à l'état de l'art pour tout un ensemble de tâches de TAL. L'adaptation de BERT au français, avec notamment le modèle CamemBERT (Martin *et al.*, 2020) a permis de répliquer ces performances pour le TAL du français. CamemBERT est basé sur RoBERTa (Liu *et al.*, 2019), une version plus efficiente de BERT. Il est entraîné sur un corpus français extrait du web nommé OSCAR (Ortiz Suárez *et al.*, 2019).

Pour extraire les informations des comptes-rendus, il est nécessaire d'avoir des modèles de langue performants sur des données cliniques françaises, notamment pour de la reconnaissance d'entités nommées. Il est possible de simplement utiliser CamemBERT, cependant les résultats de ce modèle sur des données biomédicales sont décevants (Cardon *et al.*, 2020) car sur certains jeux d'évaluation il présente des performances inférieures à des modèles heuristiques. Ces résultats sont prévisibles car il s'agit d'un modèle entraîné pour du langage courant, souvent issu de pages web de type forum, or les données biomédicales et particulièrement les données cliniques sont bien différentes. Elles présentent des termes techniques, très rares voir absents du langage courant, et un style radicalement distinct, souvent télégraphique, présentant rarement des phrases complètes avec des abréviations qui peuvent varier.

Confidentialité des données cliniques Une des problématiques majeures avec les entrepôts de données de santé est la confidentialité des données. En effet ce sont des données réglementées, soumises à des régulations par la CNIL. C'est pourquoi les adaptations de CamemBERT au domaine biomédical réalisées au sein des infrastructures des hôpitaux (Dura *et al.*, 2022) ne peuvent être publiées. Leurs jeux de données d'entraînement sont soumis à des contraintes de publication. Ces contraintes s'appliquent également aux modèles résultants. Il n'est donc pas possible d'échanger ces modèles entre différents établissements de santé. Un modèle public n'aurait pas ses contraintes, et pourrait donc être utilisé dans différents établissements.

Au travers de cet article, nous présentons deux contributions principales [1] :

— La création d'un nouveau jeu de données français public spécialisé dans le domaine biomédical
— L'introduction d'une adaptation de CamemBERT publique pour le domaine biomédical, présentant un gain de performance sur des tâches de reconnaissance d'entités nommées.

2 État de l'art

Les travaux sur l'adaptation de modèles de langue à de nouveaux domaines sont nombreux. Gururangan *et al.* (2020) montrent qu'une seconde phase de pré-entraînement sur un domaine cible

1. Nos contributions sont disponibles sur le hub Huggingface : almanach/camembert-bio-base, rntc/biomed-fr

permet d'améliorer les performances sur différentes tâches par la suite, même lorsque le corpus du domaine cible est de taille restreinte. On observe jusqu'à 3 points de gain de F-mesure dans le biomédical par rapport au même modèle sans la seconde phase.

Cet article a inspiré la création de nouveaux modèles basés sur BERT, en utilisant une seconde phase de pré-entraînement sur différents domaines spécialisés. Lee *et al.* (2019) introduit BioBERT, un modèle de type BERT spécialisé pour le biomédical anglais. BioBERT montre un gain de performance sur de nombreuses tâches de TAL biomédicales, dont $0,62\%$ d'amélioration de F-mesure sur de la reconnaissance d'entités nommées, $2,80\%$ de F-mesure sur de l'extraction de relations, et $12,24\%$ de MMR sur des questions-réponses. La seconde phase se déroule sur un corpus extrait de PubMed et PMC, composé d'articles scientifiques biomédicaux d'environ 18 milliards de mots. C'est un corpus conséquent mais toutefois composé uniquement du style scientifique. On observe cependant des gains de performance dans tous les styles. La présence du vocabulaire médical dans le corpus permet probablement une grande amélioration par rapport au langage courant.

Il est également possible d'entraîner de nouveaux modèles *from scratch*. C'est l'approche explorée par SciBERT (Beltagy *et al.*, 2019) et PubMedBERT (Gu *et al.*, 2022), deux modèles spécialisés sur des articles scientifiques biomédicaux. PubMedBERT montre que cette méthode permet de meilleures performances que les modèles entraînés par une seconde phase de spécialisation. Cependant, les gains sont assez faibles, et cette approche est particulièrement plus coûteuse. Partir de zéro nécessite un entraînement plus long et un plus grand corpus pour obtenir ces performances.

Pour le français, les modèles de référence sont CamemBERT (Martin *et al.*, 2020) et FlauBERT (Le *et al.*, 2020), cependant il n'existe pas de version biomédicale publique et disponible à ce jour de ces modèles. Cela dit, de nombreux travaux ont tenté d'adapter CamemBERT à ce domaine : Copara *et al.* (2020) ont exploré une seconde phase de pré-entraînement sur 31 000 articles scientifiques français biomédicaux. Ils n'observent néanmoins pas, sur la version large de CamemBERT, une amélioration significative sur une tâche de reconnaissance d'entités nommées cliniques. La combinaison d'un corpus assez restreint (31k documents contre 18 milliards de mots pour BioBERT) et de la version large du modèle CamemBERT, en sont probablement la cause. Le Clercq de Lannoy *et al.* (2022) ont également adapté CamemBERT au domaine biomédical. Ils ont pour cela agrégé des documents de différentes sources, tels que PubMed, Cochrane, ISTEX ou encore Wikipédia. Cela forme ainsi un plus grand corpus, partiellement public, d'environ 136 millions de mots. Ils observent une amélioration de 2 points de F-mesure sur un jeu d'évaluation de reconnaissance d'entités nommées composé de notices de médicaments (EMEA), mais pas d'amélioration significative sur un jeu composé de titres d'articles scientifiques (MEDLINE). Enfin, Dura *et al.* (2022) ont continué le pré-entraînement de CamemBERT sur 21 millions de documents cliniques de l'entrepôt de données de santé de l'APHP. On observe une amélioration significative de 3% sur APMed, un jeu de reconnaissance d'entités nommées cliniques privé appartenant à l'APHP. On note également des scores similaires à CamemBERT sur EMEA et MEDLINE. Ainsi leur nouveau modèle est meilleur sur des données cliniques et obtient des scores similaires à ceux de CamemBERT sur le reste du biomédical. Aucun de ces modèles adaptés pour le biomédical n'a été rendu public [2].

2. Un article publié suite à la soumission de notre travail annonce la publication d'un nouveau modèle biomédical français public nommé DrBERT (Labrak *et al.*, 2023)

3 Méthodes

3.1 Corpus : biomed-fr

Tout d'abord, nous avons constitué un corpus français biomédical, composé uniquement de documents publics pour minimiser les contraintes d'usage précédemment évoquées. Les documents proviennent de trois sources différentes (cf. table 1), dont la principale est ISTEX. Ce nouveau corpus que nous nommons *biomed-fr* est composé de 413 millions de mots, soit 2,7 GB de données. Martin *et al.* (2020) ont montré qu'avec seulement 4 GB de données, il était possible de quasiment égaler les performances du modèle entraîné avec les 138 GB d'OSCAR (Ortiz Suárez *et al.*, 2019). Pour une adaptation de CamemBERT au biomédical, on peut estimer que c'est une quantité de données suffisante.

Corpus	Détails	Taille
ISTEX	Divers documents de la littérature scientifique indexés sur ISTEX	276 M
CLEAR	Notices de médicaments	73 M
E3C	Divers documents issus de journaux, de notices et de cas cliniques	64 M
Total		413 M

TABLE 1 – Composition du corpus biomed-fr (en millions de mots)

ISTEX La base de données ISTEX référence 27 millions de publications scientifiques. Nous avons extrait 108 183 documents français publiés dans une revue de biologie ou de médecine depuis 1990. Les articles dont l'année de parution est plus ancienne que cette date présentent de nombreuses erreurs typographiques. Ce sont souvent des articles scannés, pour lesquels il faut appliquer des algorithmes de reconnaissance de caractères, ce qui amène à un certain nombre d'erreurs. Ce genre d'erreur se retrouve plus marginalement dans les articles publiés après 1990. Certains documents, bien qu'en français, contiennent des passages en anglais. Il y a donc une quantité indéterminée d'anglais dans ce corpus. Il est cependant peu probable que cela impacte significativement le pré-entraînement. Les erreurs typographiques et la présence d'autres langues sont des points qui pourront être corrigés par de futures versions de biomed-fr.

CLEAR Le corpus CLEAR (Grabar & Cardon, 2018) est composé d'articles d'encyclopédies, de notices de médicaments, et de résumés d'articles scientifiques. Chaque document est présent en deux versions, l'une en langage technique et l'autre en langage simplifié. Nous avons récupéré l'ensemble de ces documents dans les deux versions. Concernant les notices de médicaments, nous avons retiré les phrases redondantes en début et fin de document. Il s'agit notamment de la barre de navigation du site web dont ont été extraits les documents, ou encore d'informations sur l'entreprise qui met en vente les documents.

E3C Ce corpus (Magnini *et al.*, 2021) est composé de 3 couches. Les deux premières sont annotées ou semi-annotées, et seront retenues pour l'évaluation. La dernière couche n'est pas annotée et c'est celle que nous avons récupérée. Elle est composée de concours d'admission en spécialité de médecine,

de notices de médicaments et de résumés de thèses de médecine. Il est possible que des notices soient des doublons de ceux présents dans CLEAR.

biomed-fr-small Nous avons créé un second corpus plus petit, nommé *biomed-fr-small*. Il est constitué de 10% du contenu de biomed-fr, avec une sélection aléatoire des documents. Il va nous permettre de mesurer l'impact de la taille du corpus.

3.2 Pré-entraînements

Pour l'adaptation de CamemBERT au domaine biomédical, nous avons réalisé une seconde phase de pré-entraînement sur les deux versions du corpus biomed-fr en partant des poids et de la configuration du modèle camembert-base. Ainsi, nous avons appliqué la tâche de *Masked Language Modeling* (MLM) avec un masquage de mots entiers, pour suivre la méthode de Martin *et al.* (2020). Nous avons utilisé l'optimisateur Adam (Kingma & Ba, 2017) avec $\beta_1 = 0.9$ et $\beta_2 = 0,98$ et un taux d'apprentissage de $5e{-}5$. Nous avons effectué 50 000 pas (*steps*) pendant 39 heures avec deux Tesla V100. Nous avons utilisé une taille de lots (*batch size*) de 8 par GPU et de l'accumulation de gradient sur 16 pas pour obtenir une taille de lots effective de 256.

3.3 Affinages et évaluations

Concernant l'évaluation des modèles, nous avons récolté trois jeux de données d'évaluation de reconnaissance d'entités nommées. Les trois présentent des styles variés, ce qui permet d'évaluer la polyvalence du modèle sur les différents sous-domaines du biomédical.

QUAERO Le corpus QUAERO (Névéol *et al.*, 2014) est composé de deux jeux d'évaluation : EMEA, contenant des notices de médicaments et MEDLINE, contenant des titres d'articles scientifiques. Les entités sont annotées manuellement en suivant 10 groupes sémantiques de l'UMLS (Lindberg *et al.*, 1993). Certaines de ces entités étant imbriquées, nous avons simplement gardé les entités de plus large granularité. Les F-mesures sont calculées de la même manière.

E3C Pour l'évaluation, contrairement au corpus biomed-fr, nous utilisons les couches 1 et 2. Ces dernières présentent des documents de natures différentes. Il s'agit de cas cliniques extraits d'articles scientifiques. La couche 2 est semi-annotée. C'est celle-ci que nous utilisons comme jeu d'entraînement pour l'affinage, avec 10% dédié au jeu de validation. Nous évaluons sur la couche 1, qui est entièrement annotée à la main. Il n'y a qu'une seule classe, l'objectif est de trouver les entités cliniques dans le texte, quel que soit le type.

CAS Le corpus CAS (Grouin *et al.*, 2019) est également composé de cas cliniques issus d'articles scientifiques. Nous nous focalisons sur la tâche 3 de DEFT 2020 (Cardon *et al.*, 2020). C'est une tâche d'extraction d'information basée sur CAS. Elle comprend deux sous-tâches, et donc deux jeux d'annotations. Dans la première il faut identifier deux classes : pathologie et signe ou symptômes. La seconde concerne les informations associées : anatomie, dose, examen, mode, moment, substance, traitement et valeur. Ces deux tâches seront respectivement désignées par la suite par CAS1 et CAS2.

Affinage Concernant l'affinage, nous avons utilisé Optuna (Akiba *et al.*, 2019) pour la sélection des hyperparamètres. Ainsi nous avons un taux d'apprentissage de $5e{-}5$, un ratio d'échauffement (*warmup ratio*) de 0,224 et une taille de lots (*batch size*) de 16. Nous effectuons 2000 pas (*steps*). Les prédictions sont faites avec une simple couche linéaire en tête du modèle. Aucune des couches de CamemBERT n'est figée.

Evaluation Les scores sont mesurés avec l'outil seqeval (Nakayama, 2018) en mode strict avec micro-moyenne et le schéma "IOB2". Pour chaque évaluation, le meilleur modèle de l'affinage sur le jeu de validation est choisi pour mesurer le score final sur le jeu de test. Nous faisons la moyenne sur 10 évaluations avec différentes amorces (*seed*).

4 Résultats et discussions

				CamemBERT-bio	
Style	Dataset	Score	CamemBERT	biomed-fr-small	biomed-fr
Clinique	CAS1	F1	70,50 ± 1,75	72,94 ± 1,12	**73,03 ± 1,29**
		P	70,12 ± 1,93	**72,97 ± 0,84**	71,71 ± 1,61
		R	70,89 ± 1,78	72,92 ± 1,39	**74,42 ± 1,49**
	CAS2	F1	79,02 ± 0,92	80,00 ± 0,32	**81,66 ± 0,59**
		P	77,3 ± 1,36	78,29 ± 0,91	**80,96 ± 0,91**
		R	80,83 ± 0,96	81,80 ± 0,48	**82,37 ± 0,69**
	E3C	F1	67,63 ± 1,45	67,96 ± 1,85	**69,85 ± 1,58**
		P	78,19 ± 0,72	77,41 ± 1,01	**79,11 ± 0,42**
		R	59,61 ± 2,25	60,57 ± 2,32	**62,56 ± 2,50**
Notices	EMEA	F1	74,14 ± 1,95	75,93 ± 2,42	**76,71 ± 1,50**
		P	74,62 ± 1,97	76,23 ± 2,27	**76,92 ± 1,96**
		R	73,68 ± 2,22	75,63 ± 2,61	**76,52 ± 1,62**
Scientifique	MEDLINE	F1	65,73 ± 0,40	65,48 ± 0,31	**68,47 ± 0,54**
		P	64,94 ± 0,82	64,43 ± 0,50	**67,77 ± 0,88**
		R	66,56 ± 0,56	66,56 ± 0,16	**69,21 ± 1,32**

TABLE 2 – Moyennes sur 10 évaluations des F-mesures sur différents jeux biomédicaux de reconnaissance d'entités nommées

CamemBERT vs CamemBERT-bio On observe (cf. table 2) un gain significatif de performance sur tous les jeux d'évaluation avec notre nouveau modèle. Nous avons en moyenne 2,54 points d'amélioration de F-mesure. Ce gain s'observe dans tous les styles, ce qui montre la polyvalence du modèle pour les domaines cliniques et scientifiques du biomédical.

biomed-fr-small vs biomed-fr On note une baisse de performance avec le jeu biomed-fr-small, mais toujours un gain significatif sur certains jeux de données par rapport à CamemBERT. Cela

confirme que la taille du corpus influe positivement sur les performances même dans un domaine spécialisé comme le biomédical.

Évaluateur	Auteurs	CAS1	CAS2	EMEA			MEDLINE		
		F1	F1	F1	P	R	F1	P	R
seqeval	Dura *et al.* (2022)-fine-tuned	-	-	72,90	-	-	59,70	-	-
	Dura *et al.* (2022)-from-scratch	-	-	69,30	-	-	60,10	-	-
	Notre approche	**73,03**	**81,66**	**76,71**	**76,92**	**76,52**	**68,47**	**67,77**	**69,21**
BRATeval	Le Clercq de Lannoy *et al.* (2022)			67,4	73,4	62,2	55,3	62,2	49,7
	Mulligen *et al.* (2016)	-	-	74,9	71,6	**78,5**	**69,8**	68	**71,6**
	Copara *et al.* (2020)	61,53	73,7	-	-	-	-	-	-
	Notre approche	**84,97**	**83,25**	**77,80**	**79,77**	75,93	56,16	**75,33**	44,82

TABLE 3 – Comparaison de CamemBERT-bio avec différentes approches sur les 4 tâches de reconnaissance d'entités nommées. Dans la première partie de la table les scores sont mesurés avec seqeval (Nakayama, 2018), et la seconde avec BRATeval, qui est l'outil d'évaluation fourni pour la campagne CLEF eHealth Evaluation lab 2016 (Névéol *et al.*, 2016).

Auteurs	Corpus d'adaptation	
	Origine	Taille
Dura *et al.* (2022)	APHP	21 MD
Le Clercq de Lannoy *et al.* (2022)	divers	136 MW
Copara *et al.* (2020)	PubMed	31 KD
Notre approche	biomed-fr	413 MW

TABLE 4 – Corpus de pré-entraînement des différents approches (cf. table 3)

Comparaison avec l'état de l'art Nous avons comparé les performances de CamemBERT-bio avec les différentes approches précédemment évoquées (cf. table 3). CamemBERT-bio obtient pour presque tous les jeux d'évaluation les meilleurs résultats. Dura *et al.* (2022) n'ont pas observé d'amélioration sur EMEA et MEDLINE par rapport à CamemBERT car leur corpus de pré-entraînement (cf. table 4) est composé de documents provenant de l'APHP, ce qui en fait un corpus moins varié. Ils gagnent cependant plusieurs points sur leur jeu d'évaluation basé lui aussi sur les documents de l'APHP. Mulligen *et al.* (2016) présente le meilleur score sur MEDLINE, ainsi que le meilleur rappel sur EMEA. Leur approche est basée sur un modèle à base de connaissances, ce qui leur permet d'obtenir le meilleur rappel sur les deux jeux d'évaluations de QUAERO. Enfin, leur approche est la seule capable de gérer les entités imbriquées, ce qui leur donne un avantage.

Il est important de noter que ces différentes approches basées sur CamemBERT ont des cadres expérimentaux variés. La présence de couches CRF plutôt qu'une simple couche linéaire en sortie de CamemBERT, le gel des couches de CamemBERT ou encore les hyperparamètres sont des exemples de variation qu'on observe en plus des corpus de pré-entraînement, ce qui rend la comparaison plus difficile.

Analyse de la tokenisation CamemBERT-bio est un modèle adapté pour le biomédical de CamemBERT. Contrairement à un nouveau modèle entraîné *from scratch*, il partage le même vocabulaire. Le vocabulaire de CamemBERT a été construit en utilisant SentencePiece (Kudo & Richardson, 2018) sur un échantillon d'OSCAR. C'est donc un vocabulaire généraliste fait pour le langage courant. On peut faire l'hypothèse que le tokeniseur de CamemBERT va produire des sur-segmentations de termes techniques biomédicaux.

Nous avons alors exploré cette possibilité en entraînant un tokeniseur spécialisé sur biomed-fr-small, et en calculant l'intersection des deux vocabulaires.

Termes	généraliste	spécialisé
échocardiographie	écho-cardi-ographie	échocardiographi-e
transthoracique	trans-thorac-ique	trans-thoracique
glimépiride	g-lim-épi-ride	gli-m-épi-ride
cardiopathie	cardio-pathie	cardiopathie
diastoliques	dia-s-tol-iques	diastolique-s

TABLE 5 – Comparaison de la segmentation entre un tokeniseur généraliste et un tokeniseur spécialisé sur quelques termes techniques biomédicaux

On calcule une intersection de 45% entre les deux vocabulaires, ce qui est assez proche de l'intersection de 42% trouvé par Beltagy *et al.* (2019) entre le vocabulaire de BERT et celui de SciBERT. Il y a donc une différence significative des termes les plus fréquents.

5 Conclusion et perspectives

Nous avons introduit un nouveau corpus biomédical français nommé *biomed-fr* de 413 millions de mots composé de notices de médicaments et de documents de la littérature scientifique en médecine et en biologie. Ce nouveau corpus nous a permis d'adapter CamemBERT au domaine biomédical avec une seconde phase de pré-entraînement. On observe une amélioration des performances sur tous nos jeux d'évaluation de reconnaissance d'entités nommées. On note un gain de 2,54 points de F-mesure en moyenne.

Nous avons des pistes pour de nouvelles versions de biomed-fr. D'une part, réaliser un plus grand nettoyage des données en retirant les passages au sein des documents qui ne sont pas en français, ou en retirant les documents contenant un trop grand nombre d'erreurs typographiques. D'autre part, en augmentant la quantité de données. Cela pourrait passer par l'exploitation des documents archivés sur HAL concernant les sciences de la vie, notamment publiés par l'INSERM, ou la récupération de résumés d'articles français sur PubMed.

L'analyse de la tokenisation nous pousse à réfléchir à la création d'un nouveau vocabulaire pour CamemBERT-bio. Malgré le gain de performance assez faible de PubMedBERT par rapport à BioBERT malgré son vocabulaire spécialisé, la sur-segmentation des termes techniques et le faible taux d'intersection entre le vocabulaire généraliste et le vocabulaire spécialisé montrent l'intérêt de

l'expérience.

Enfin, ces derniers mois, de nombreux modèles génératifs, souvent de plusieurs milliards de paramètres, ont montré des performances remarquables sur des tâches biomédicales, outrepassant parfois les modèles spécialisés comme BioBERT (Agrawal *et al.*, 2022; Singhal *et al.*, 2022). C'est une piste de recherche prometteuse pour l'extraction d'information biomédicale. Cependant nous avons des raisons de penser qu'un modèle de type BERT a toujours de l'intérêt (Lehman *et al.*, 2023). D'une part, dans le contexte clinique, les modèles doivent souvent être utilisés au sein des infrastructures des établissements de santé, ce qui se traduit par des contraintes de ressources. Il est alors plus facile de déployer des petits modèles spécialisés que des grands modèles généralistes. D'autre part l'utilisation de ces modèles génératifs nécessite souvent de passer par des serveurs distants, souvent à travers des API, ce qui rend difficile leur utilisation compte tenu des contraintes de confidentialité auxquelles sont soumis les documents cliniques.

Références

AGRAWAL M., HEGSELMANN S., LANG H., KIM Y. & SONTAG D. (2022). Large Language Models are Few-Shot Clinical Information Extractors. arXiv :2205.12689 [cs], DOI : 10.48550/arXiv.2205.12689.

AKIBA T., SANO S., YANASE T., OHTA T. & KOYAMA M. (2019). Optuna : A Next-generation Hyperparameter Optimization Framework. arXiv :1907.10902 [cs, stat], DOI : 10.48550/arXiv.1907.10902.

BELTAGY I., LO K. & COHAN A. (2019). SciBERT : A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3615–3620, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1371.

BENAMARA F., HATOUT N., MULLER P. & OZDOWSKA S., Éds. (2007). *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse. ATALA, IRIT.

CARDON R., GRABAR N., GROUIN C. & HAMON T. (2020). Présentation de la campagne d'évaluation DEFT 2020 : similarité textuelle en domaine ouvert et extraction d'information précise dans des cas cliniques. In *DEFT 2020*, Nancy, France.

COPARA J., KNAFOU J., NADERI N., MORO C., RUCH P. & TEODORO D. (2020). Contextualized French Language Models for Biomedical Named Entity Recognition. In R. CARDON, N. GRABAR, C. GROUIN & T. HAMON, Éds., *Traitement Automatique des Langues Naturelles (TALN, 27e édition). Atelier DÉfi Fouille de Textes*, p. 36–48, Nancy, France : ATALA.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv :1810.04805 [cs], DOI : 10.48550/arXiv.1810.04805.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

DURA B., JEAN C., TANNIER X., CALLIGER A., BEY R., NEURAZ A. & FLICOTEAUX R. (2022). *Learning structures of the French clinical language :development and validation of word embedding models using 21 million clinical reports from electronic health records*. Rapport interne arXiv :2207.12940, arXiv. arXiv :2207.12940 [cs, stat] type : article.

GRABAR N. & CARDON R. (2018). CLEAR – Simple Corpus for Medical French. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, p. 3–9, Tilburg, the Netherlands : Association for Computational Linguistics. DOI : 10.18653/v1/W18-7002.

GROUIN C., GRABAR N., CLAVEAU V. & HAMON T. (2019). Clinical Case Reports for NLP. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, p. 273–282, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-5029.

GU Y., TINN R., CHENG H., LUCAS M., USUYAMA N., LIU X., NAUMANN T., GAO J. & POON H. (2022). Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare*, **3**(1), 1–23. arXiv :2007.15779 [cs], DOI : 10.1145/3458754.

GURURANGAN S., MARASOVIĆ A., SWAYAMDIPTA S., LO K., BELTAGY I., DOWNEY D. & SMITH N. A. (2020). Don't Stop Pretraining : Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8342–8360, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.740.

KINGMA D. P. & BA J. (2017). Adam : A Method for Stochastic Optimization. arXiv :1412.6980 [cs], DOI : 10.48550/arXiv.1412.6980.

KUDO T. & RICHARDSON J. (2018). SentencePiece : A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 66–71, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-2012.

LABRAK Y., BAZOGE A., DUFOUR R., ROUVIER M., MORIN E., DAILLE B. & GOURRAUD P.-A. (2023). Drbert : A robust pre-trained model in french for biomedical and clinical domains.

LAIGNELET M. & RIOULT F. (2009). Repérer automatiquement les segments obsolescents à l'aide d'indices sémantiques et discursifs. In A. NAZARENKO & T. POIBEAU, Éds., *Actes de TALN 2009 (Traitement automatique des langues naturelles)*, Senlis : ATALA LIPN.

LANGLAIS P. & PATRY A. (2007). Enrichissement d'un lexique bilingue par analogie. In (Benamara *et al.*, 2007), p. 101–110.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020). Flaubert : Unsupervised language model pre-training for french.

LE CLERCQ DE LANNOY T., BESANÇON R., FERRET O., TOURILLE J., BRIN-HENRY F. & VIERU B. (2022). Stratégies d'adaptation pour la reconnaissance d'entités médicales en français (Adaptation strategies for biomedical named entity recognition in French). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 215–225, Avignon, France : ATALA.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2019). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, p. btz682. arXiv :1901.08746 [cs], DOI : 10.1093/bioinformatics/btz682.

LEHMAN E., HERNANDEZ E., MAHAJAN D., WULFF J., SMITH M. J., ZIEGLER Z., NADLER D., SZOLOVITS P., JOHNSON A. & ALSENTZER E. (2023). Do We Still Need Clinical Language Models ? arXiv :2302.08091 [cs], DOI : 10.48550/arXiv.2302.08091.

LINDBERG D. A., HUMPHREYS B. L. & MCCRAY A. T. (1993). The Unified Medical Language System. *Methods of Information in Medicine*, **32**(4), 281–291. DOI : 10.1055/s-0038-1634945.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. arXiv :1907.11692 [cs], DOI : 10.48550/arXiv.1907.11692.

MAGNINI B., ALTUNA B., LAVELLI A., SPERANZA M. & ZANOLI R. (2021). The E3C Project : Collection and Annotation of a Multilingual Corpus of Clinical Cases. In J. MONTI, F. TAMBURINI & F. DELL'ORLETTA, Éds., *Proceedings of the Seventh Italian Conference on Computational Linguistics CLiC-it 2020 : Bologna, Italy, March 1-3, 2021*, Collana dell'Associazione Italiana di Linguistica Computazionale, p. 258–264. Torino : Accademia University Press. Code : Proceedings of the Seventh Italian Conference on Computational Linguistics CLiC-it 2020 : Bologna, Italy, March 1-3, 2021.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics.

MULLIGEN E. M. V., AFZAL Z., AKHONDI S. A., VO D. & KORS J. A. (2016). Erasmus MC at CLEF eHealth 2016 : Concept Recognition and Coding in French Texts.

NAKAYAMA H. (2018). seqeval : A python framework for sequence labeling evaluation. Software available from https ://github.com/chakki-works/seqeval.

NÉVÉOL A., COHEN K. B., GROUIN C., HAMON T., LAVERGNE T., KELLY L., GOEURIOT L., REY G., ROBERT A., TANNIER X. & ZWEIGENBAUM P. (2016). Clinical Information Extraction at the CLEF eHealth Evaluation lab 2016. *CEUR workshop proceedings*, **1609**, 28–42.

NÉVÉOL A., GROUIN C., LEIXA J., ROSSET S. & ZWEIGENBAUM P. (2014). The QUAERO French medical corpus : A ressource for medical entity recognition and normalization. In *Proc of BioTextMining Work*, p. 24–30.

ORTIZ SUÁREZ P. J., SAGOT B. & ROMARY L. (2019). Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, p. 9 – 16, Mannheim : Leibniz-Institut für Deutsche Sprache. DOI : 10.14618/ids-pub-9021.

RAGHAVAN P., CHEN J. L., FOSLER-LUSSIER E. & LAI A. M. (2014). How essential are unstructured clinical narratives and information fusion to clinical trial recruitment ? *AMIA Summits on Translational Science Proceedings*, **2014**, 218–223.

SERETAN V. & WEHRLI E. (2007). Collocation translation based on sentence alignment and parsing. In (Benamara *et al.*, 2007), p. 401–410.

SINGHAL K., AZIZI S., TU T., MAHDAVI S. S., *et al.* (2022). Large Language Models Encode Clinical Knowledge. arXiv :2212.13138 [cs], DOI : 10.48550/arXiv.2212.13138.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Protocole d'annotation multi-label pour une nouvelle approche à la génération de réponse socio-émotionnelle orientée-tâche

Lorraine Vanel [1, 2], Alya Yacoubi [2] Chloé Clavel [1]
(1) LTCI, Telecom-Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France
(2) Zaion, 75008 Paris, France
lorraine.vanel@telecom-paris.fr, chloé.clave@telecom-paris.fr,
ayacoubi@zaion.ai

RÉSUMÉ

Depuis l'apparition des systèmes conversationnels, la modélisation des comportements humains constitue un axe de recherche majeur afin de renforcer l'expression des attributs émotionnels de ces systèmes. En nous intéressant aux agents conversationnels génératifs orientés-tâches, nous proposons une nouvelle approche pour rendre la réponse générée plus pertinente au contexte émotionnel de l'interlocuteur. Cette approche consiste à ajouter une étape supplémentaire de prédiction de labels pour conditionner la réponse générée et assurer sa pertinence au contexte socio-émotionnel de l'utilisateur. Nous proposons une formulation de cette nouvelle tâche de prédiction en nous appuyant sur un protocole d'annotation de données que nous avons conçu et implémenté. À travers cet article, nous apportons les contributions suivantes : la formulation de la tâche de prédiction de labels socio-émotionnels et la description du protocole d'annotation associé. Avec cette méthodologie, nous visons à développer des systèmes conversationnels socialement pertinents et indépendants.

ABSTRACT

Multi-label annotation protocol for a new approach to task-oriented socio-emotional response generation

Ever since the emergence of conversational systems, modeling human behaviors has been a major research topic, aiming to improve the expression of the social and emotional attributes of these systems. With a focus on task-oriented generative conversational agents, we propose a new approach to make the generated response more relevant to the emotional context of the interlocutor. This approach consists in adding an additional label prediction step to condition the generated response and ensure its consistency to the user's socio-emotional context. We propose a formulation of this new prediction task based on a data annotation protocol that we have designed and implemented. In this paper, we introduce the following contributions : the formulation of the socio-emotional label prediction task and the description of the associated annotation protocol. With this methodology we aim at developing socially relevant and independent conversational systems.

MOTS-CLÉS : Génération de Réponse Émotionnelle ; Dialogue Social ; Système Conversationnel Socio-Émotionnel ; Prédiction Socio-Émotionnelle ; Protocole d'Annotation.

KEYWORDS: Emotional Response Generation ; Affective Dialogue ; Social Dialogue ; Socio-Emotionnal Conversational Systems ; Socio-Emotional Label Prediction ; Annotation Protocol.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 335–348.

1 Introduction

L'essor des technologies de traitement du langage a favorisé l'émergence de nombreux assistants virtuels et autres systèmes conversationnels utilisés dans le secteur de la relation client (Ram *et al.*, 2018; Gnewuch *et al.*, 2017), de la formation ou encore de la médecine (Harilal *et al.*, 2020; Adikari *et al.*, 2022). Traditionnellement basés sur des systèmes de règles ou sur une architecture modulaire composée de briques d'intelligence artificielle séparées, une nouvelle ère de l'IA se dessine : celle de la génération automatique de réponse permettant ainsi un dialogue personnalisé et naturel. Cependant, cette approche implique plusieurs défis techniques, tels que la difficulté à combiner l'aspect orienté-tâche avec la dimension émotionnelle des interactions (Clavel *et al.*, 2022). C'est cette prise en compte spontanée de l'émotion latente, naturelle dans un dialogue entre humains, que nous souhaitons apprendre à nos systèmes. Nous faisons donc l'hypothèse qu'utiliser des données conversationnelles humain-humain est l'approche optimale pour entraîner des modèles capables de détecter la réaction émotionnelle de l'interlocuteur et de prendre en considération le contexte social de l'interaction.
Dans la littérature des systèmes de génération de dialogues sociaux et émotionnels, de nombreux travaux proposent de classer les tours de paroles avec des labels comme des émotions, des actes de dialogues ou encore des stratégies de dialogue afin d'assurer la cohérence de la conversation (Li *et al.*, 2017; Welivita *et al.*, 2021). Bien que dans la majorité des cas, un unique label est donné par tour de parole, celui-ci est souvent constitué de plusieurs segments consécutifs qui présentent des stratégies émotionnelles et dialogiques différentes et logiquement dépendantes.
Dans cet article, nous présentons une nouvelle approche pour la tâche de génération de réponse socio-émotionnelle, où une séquence de labels consécutifs est prédite pour plannifier et conditionner la génération. Cela se traduit par l'ajout d'une étape de prédiction de ces labels représentant les comportements attendus dans la réponse du système. Nous proposons les contributions suivantes :

— La formulation de la tâche additionnelle de prédiction de label socio-émotionnel du prochain tour de parole.

— La description d'un protocole d'annotation d'un corpus socio-émotionnel.

Nous commençons par faire une revue des travaux effectués dans le domaine des systèmes de dialogues socio-émotionnels. Ensuite, nous formalisons la tâche de prédiction de labels socio-émotionnels pour détailler le protocole d'annotation adapté, avant d'analyser un corpus annoté selon ces directives.

2 État de l'art

2.1 Les modèles de génération de dialogue émotionnel dans la littérature

Les modèles neuronaux appris sur des corpus de grande taille ont permis d'accélérer l'essor du domaine de la génération de dialogue (Shuster *et al.*, 2022; Zhang *et al.*, 2019; Thoppilan *et al.*, 2022) Certains de ces modèles ont été utilisés pour générer des dialogues socio-émotionnels. Dans ce cas, une annotation émotionnelle des données d'apprentissage est souvent nécessaire. Rashkin *et al.* (2018) proposent un *dataset* annoté en émotions au niveau de la conversation et évalué sur un modèle Transformers. Ce label émotionnel couvre donc la conversation entière, et malgré la précision de la liste de labels utilisée, cette approche ne permet pas d'observer les comportements émotionnels au niveau du tour de parole. Kumar *et al.* (2018) montrent que l'utilisation des actes de dialogue en tant que labels augmente considérablement les performances des systèmes conversationnels. Cependant,

la dimension émotionnelle, au cœur du dialogue humain, n'est pas représentée par ces actes de dialogue. C'est ainsi que des jeux de données comme DailyDialog (Li *et al.*, 2017) (utilisé par Zandie & Mahoor (2020) dans leur modèle EmpTransfo, par exemple) ou le Emotional Dialogue in OpenSubtitles (EDOS) par Welivita *et al.* (2021), présentant une double annotation des stratégies de dialogues et des émotions, capturent les deux aspects que nous cherchons à étudier. Aussi complets soient-ils, ces jeux de données sont composés d'interactions en domaine ouvert et scriptées, ce qui biaise l'authenticité des réponses émotionnelles. De plus, ces corpus ne présentent qu'un unique label par tour de parole, ce qui ne permet pas de représenter les évolutions dynamiques au sein d'un même tour.
Nous avons donc décidé d'explorer les différentes manières d'annoter des données conversationnelles, afin de mettre au point un protocole d'annotation incluant les stratégies de dialogue et les émotions. Cette double annotation nous permet d'exprimer les changements et relations entre les stratégies conversationnelles (stratégies de dialogue et émotions).

2.2 Travaux d'annotation dans la littérature

Certains travaux se sont intéressés à la labellisation des conversations pour conditionner la réponse. Nous énumérons les approches les plus étudiées dans la littérature.

Collecte des données

— ***Crowd-sourcing*** Appliqué à la collecte de données, le *crowd-sourcing* est une méthode participative où un groupe de personnes contribue à la création d'échantillons de données. Les données collectées sont généralement des interactions humains-humain (H-H). Les données sont collectées en faisant interagir deux participants en suivant des directives précises : le locuteur met en place une situation, souvent initiée par un prompt émotionnel (Rashkin *et al.*, 2018; Liu *et al.*, 2021) et l'auditeur doit répondre en conséquence, sans connaître le prompt initial. Les systèmes de dialogue sont formés pour jouer le rôle d'auditeurs.

— **Extraction du Web** Une autre façon courante de collecter des données consiste à extraire des informations de sources en ligne. Dans le cas des données textuelles, il s'agit souvent de messages et de commentaires récupérés sur les réseaux sociaux et il s'agit donc de discours naturel entre humains (Zhong *et al.*, 2020; Mazaré *et al.*, 2018). Elles peuvent également provenir d'autres sources, comme OpenSubtitles (Welivita *et al.*, 2021) où les données sont scénarisées. Les données extraites de ces sites web ne sont généralement pas étiquetées et des processus d'annotation doivent être conçus pour annoter les corpus.

— **Enregistrements de conversations** Cette approche peut être utilisée pour récupérer des archives de conversation humain-humain à partir de données de centres d'appels, comme dans Clavel *et al.* (2013). Ces données sont moins accessibles, car cette pratique nécessite d'avoir les moyens de déployer de tels services ou de demander des données à une entreprise disposant de telles ressources. Même dans ce cas, les données sont généralement confidentielles et ne peuvent donc pas être partagées en tant que jeux de données publics, à moins que le consentement de l'utilisateur ait été donné et que les données aient été correctement anonymisées.

Annotation des données Il existe plusieurs approches de l'annotation des données : elles diffèrent selon le point de vue de l'annotateur, ou encore par les ressources nécessaires à la tâche d'étiquetage.

— **Annotation externe** L'annotation externe, ou du point de vue observateur, implique que le label est donné après analyse par un parti indépendant. Cette approche peut être utilisée sur tout type de données.
  — **Annotation manuelle** Cette approche consiste à entièrement annoter un jeu de données par des experts humains ou des annotateurs qui ont été formés à la tâche spécifique d'annotation. Li *et al.* (2017) présentent un jeu de données de 13K, DailyDialog, qui a été annoté par 3 experts ayant une bonne compréhension de la théorie du dialogue et de la communication, et qui ont été formés aux directives de la tâche particulière (c'est-à-dire l'annotation des émotions et des actes de dialogue). Toutefois, l'annotation manuelle d'un large corpus peut être très coûteuse en temps et en ressources matérielles.
  — **Annotation semi-automatique** Associer l'annotation manuelle à l'usage de modèles permet d'accélérer la tâche d'annotation et d'alléger la charge de travail des juges humains (Lu *et al.*, 2021; Welivita *et al.*, 2021). Il existe de nombreuses manières différentes de réaliser une telle annotation. En général, la première étape consiste à faire annoter par des juges humains une petite fraction des dialogues collectés. Ce sous-ensemble est ensuite utilisé pour entraîner un modèle de classification qui peut soit automatiquement annoter le reste du corpus considéré, ou proposer les labels les plus probables pour chaque exemple du corpus restant. Dans le deuxième cas, c'est aux annotateurs humains de choisir parmi les annotations proposées par le modèle de classification, pour apporter la décision finale.
— **Annotation interne** L'annotation est dite interne lorsque le label est directement dérivé de la source de la donnée.
  — ***Crowd-sourcing*** C'est la principale méthode d'annotation pour ces données, où les émotions et les étiquettes de stratégies de dialogue associées aux données peuvent être directement dérivées des instructions données aux annotateurs (Rashkin *et al.*, 2018; Liu *et al.*, 2021). De plus, Liu *et al.* (2021) recueillent les réponses aux enquêtes soumises aux participants pendant le processus de collecte, tant du côté de l'auditeur que du locuteur. Cela permet de recueillir davantage de données telles que la notation de l'empathie et les stratégies de dialogue au niveau de l'énoncé.
  — **Données extraites d'internet** Zhong *et al.* (2020) utilisent le contexte dans lequel les données web ont été postées et extraient les messages et les commentaires sur deux subreddits : `/r/happy` et `/r/offmychest`. L'environnement original de Reddit fournit donc une étiquette et ce qu'il reste à faire est un contrôle de qualité en demandant à des annotateurs humains d'annoter un petit ensemble de conversations : 100 du Reddit `/r/happy`, 100 de `/r/offmychest` et pour le contrôle, 100 de `/r/casualconversations`.
  — **Enregistrements de conversation** Dans ce cas, l'annotation peut venir d'un retour de l'utilisateur. En effet, certains bots déployés demandent un retour sur la satisfaction des clients, soit directement, soit par le biais de sondages. Ces informations peuvent être utilisées pour annoter certaines conversations (Maslowski *et al.*, 2017; Guibon *et al.*, 2021).

La différence de point de vue de l'annotateur est particulièrement marquée lorsque les labels considérés sont aussi subjectifs que l'émotion. L'annotateur interne peut étiqueter avec précision l'émotion qu'il ressent et exprime dans les données, même si cela vient souvent au prix d'interactions jouées et scénarisées. Un annotateur externe fait une hypothèse quant à l'émotion exprimée, et sa perception est colorée par ses expériences culturelles et ses sensibilités personnelles. Cependant, cette approche est accessible et peut être réalisée sur de nombreux formats de données, ce qui laisse la possibilité d'annoter, à posteriori, des données humain-humain spontanées. Il faut donc garder à l'esprit que

les deux approches comportent leurs propre biais, que ce soit au niveau de la nature scriptée des interaction ou du biais de l'annotation en elle-même. Cette subjectivité de la tâche impacte également les attentes au niveau des scores inter-annotateurs attendus.

2.3 Stratégies et Labels Socio-Émotionnels

Nous présentons enfin les différentes stratégies utilisées dans les approches de génération conversationnelle socio-émotionnelle, ainsi que les différentes manières dont celles-ci sont représentées en tant que labels dans les différents jeux de données disponible dans la littérature. Les jeux de données que nous citons sont en anglais, car nous n'avons trouvé que peu de ressources conversationnelles et annotées en labels sociaux ou émotionnels en français. Nous rappelons que dans cette étude, nous définissons la notion de dialogue "social" par les aspects relationnels liés aux différentes attitudes sociale et à la communication inter-personnelle.

Stratégies basées sur les émotions

— **Définition** Les stratégies basées sur les émotions font référence aux approches qui relèvent de la détection, du traitement et de l'expression d'une émotion en réponse à une situation émotionnelle de l'utilisateur. L'une des approches émotionnelles les plus représentées dans la littérature est l'usage de l'empathie (Fung *et al.*, 2018; Wang *et al.*, 2021; Hosseini & Caragea, 2021), définie comme la capacité à se mettre à la place de son interlocuteur et de percevoir ce qu'iel ressent (Cuff *et al.*, 2016).

— **Labels** Il existe plusieurs manières d'annoter les émotions. Le premier niveau est le sentiment, en annotant la polarité positive ou négative (Lu *et al.*, 2021). Pour ce qui est des émotions plus fines, de nombreuses études font référence à diverses théories de la psychologie pour la classification des émotions, mais il n'y a pas de consensus sur la manière de définir et de classer les émotions dans l'analyse des conversations (Clavel & Callejas, 2016). Li *et al.* (2017) basent leurs annotations sur le modèle d'Ekman (Ekman, 1999), et Liu *et al.* (2021); Rashkin *et al.* (2018) utilisent des théories classiques dérivées des réponses biologiques (Ekman, 1999; Plutchik, 1984) ainsi que des études concentrées sur un ensemble plus large d'émotions subtiles et dépendantes du contexte (Skerry *et al.*, 2015) atteignant jusqu'à 32 étiquettes d'émotions. (Feng *et al.*, 2021) utilisent une classification adaptée aux contextes orientés-tâche.

Stratégies de dialogue

— **Définition** Les stratégies de dialogues sont un ensemble d'actions conversationnelles qui permettent d'exprimer une intention conversationnelle (Galescu *et al.*, 2018; Santos Teixeira & Dragoni, 2022; Liu *et al.*, 2021). Certaines stratégies de dialogue telles qu'*informer* ou *questionner*, se rapprochent des actes de dialogues qui peuvent être interprétés comme la réalisation de ces stratégies. D'autres sont plus émotives, comme *sympathiser* ou *encourager* (Welivita & Pu, 2020).

— **Labels** Certains jeux de données sont doublement annotés en émotions et en stratégies de dialogue. Les deux principales approches que nous avons vues pour ces stratégies sont les stratégies de dialogues elles-mêmes (Hardy *et al.*, 2021; Welivita & Pu, 2020; Liu *et al.*, 2021). Elles peuvent également être associées aux actes de dialogue, résumés et classifiés dans les travaux de Bunt (2006), comme dans le *dataset* DailyDialog (Li *et al.*, 2017).

Stratégies de conception de persona

— **Définition** Une persona est une personnalité fictive dotée de caractéristiques sociales, comme des traits de personnalité ou des préférences (Li *et al.*, 2016). Associer une persona à un système conversationnel revient alors à conditionner les réponses de l'agent pour prendre en compte ces attributs, ce qui permet d'unifier et d'améliorer la cohérence du comportement du système. Plusieurs études relatent les différentes stratégies de conception de ces personas, et de l'influence de certaines caractéristiques sur l'accueil et l'acceptabilité des systèmes chez les utilisateurs (Pradhan & Lazar, 2021; Kim *et al.*, 2019).

— **Labels** Les corpus définissent généralement une persona comme un ensemble de phrases, représentant la personnalité choisie, sur laquelle vont se baser la formulation et le comportement général de l'agent (Mazaré *et al.*, 2018; Zhang *et al.*, 2018). Dans le cadre de données conversationnelles, ces phrases sont souvent collectées à partir de profils d'utilisateurs sur internet, un utilisateur représentant une persona (Zhong *et al.*, 2020; Mazaré *et al.*, 2018).

Ainsi, de nombreuses études s'intéressent à la tâche de génération neuronale de réponse sociale et émotionnelle, en basant leurs approches respectives sur des corpus adéquats. La manière dont ces données sont collectées et annotées en émotions et en stratégies de dialogue est cruciale à la réalisation de la tâche. Le plus souvent, l'annotation est à l'échelle du tour de parole, mais nous souhaitons viser une unité plus fine, en segmentant le tour de parole en segments. Nous nous inspirons des travaux cités pour proposer notre méthodologie d'annotation qui répond au défi technologique que nous souhaitons adresser : la prise en compte dynamique de l'évolution des états émotionnels et dialogiques au cours d'une conversation orientée-tâche.

3 Notre proposition d'annotation multi-label des conversations pour un système de dialogue génératif socio-émotionnel

3.1 Formalisation de la tâche d'annotation en séquence de labels socio-émotionnel

Soit une conversation $C = (c_i)_{i\in[0,t]}$, c_t le tour de parole en cours. Soit SE la liste des labels socio-émotionnels considérés. $\forall i \in [0,t], \exists y_i = (y_i^j)_{j\in[0,l_i]}$, une séquence ordonnée de labels socio-émotionnels associés au tour de parole c_i et $l_i \in \mathbb{N}$ le nombre de labels associés à c_i. $\forall i \in [0,t], \forall j \in [0,l_i], y_i^j \in SE$. Nous considérons maintenant la tâche de prédiction de la séquence y_{t+1} des labels socio-émotionnels associés au tour de parole $t+1$, c_{t+1}. En d'autre termes, il s'agit de prédire l'ensemble :

$$y_{t+1} = (y_{t+1}^j)_{j\in[0,l_{t+1}]} \in SE^{l_{t+1}}$$

Cette séquence sera ensuite utilisée afin d'influencer le système, en prenant en compte les comportement désirés représentés par ces labels pour les générer dans le tour suivant.

3.2 Conception du protocole d'annotation

Dans le but de développer un système pour accomplir la tâche formulée ci-dessus, nous avons conçu un protocole spécifique pour guider la construction d'un corpus annoté avec ces labels socio-émotionnels. Tout d'abord, nous avons recueilli nos données à partir d'enregistrements de conversations entre clients et agents, extraites de nos systèmes déployés dans le cadre industriel. Notre objectif est donc d'annoter manuellement ces conversations avec des labels émotionnels et des stratégies de dialogue. Ce corpus sera utilisé par la suite pour entraîner notre modèle de génération de réponses socialement et émotionnellement pertinentes. Notre jeu de données est composé de 72 conversations téléphoniques en français qui ont été transcrites manuellement par cinq experts en analyse linguistique. Le caractère personnel de ces informations, recueillies en accord avec les régulations RGPD et le consentement des clients, ne nous permets pas de partager ce *dataset*. Cette même équipe d'analystes a également mené la tâche d'annotation de ce corpus textuel.

3.2.1 Préparation des données

Après une revue approfondie de la littérature des systèmes de dialogues socio-émotionnels présentée dans nos travaux précédents (Vanel. *et al.*, 2023), nous avons composé une première liste d'étiquettes pour les émotions et les stratégies de dialogue. Nous avons ensuite analysé nos données pour sélectionner les étiquettes pertinentes. Pour ce faire, nous avons annoté un échantillon de conversations avec la liste complète pour noter les labels manquants ou superflus. Par exemple, dans cette étape, nous avons ajouté la stratégie de politesse, pour indiquer les formules de politesse comme les salutations et les remerciements. Après cette étape, nous avons pu établir la liste finale SE des labels socio-émotionnels. Ces labels ont été annotés à deux niveaux : au niveau du tour de parole (stratégies de dialogue et émotions) et au niveau de la conversation (indices de satisfaction globale). Nous décrivons ces tâches d'annotation plus en détail ci-dessous.

3.2.2 Tâches d'étiquetage

Au niveau du tour de parole Comme indiqué sur la Figure 1, nous cherchons à annoter une liste de labels, issue de la fusion des annotations issues des deux tâches suivantes :

— **Émotions** Pour cette tâche, nous demandons aux analystes d'étiqueter les émotions exprimées à chaque tour de conversation. Les experts lisent chaque tour de conversation, en analysant leur contenu sémantique. Si une émotion est détectée chez un locuteur, l'annotateur note l'émotion et le segment textuel porteur de l'émotion. À la fin de cette tâche, toutes les conversations sont étiquetées en émotions et leurs "indices sémantiques", au niveau du tour de parole.

— **Stratégies de dialogue** Pour les stratégies de dialogue, nous divisons l'annotation en deux tâches : l'annotation des tours de l'agent et l'annotation des tours du client. Ces tâches sont similaires, mais elles sont conduites séparément. Chaque stratégie de dialogue a un code, par exemple la stratégie "Information" est codée par la lettre "I". Un analyste passe sur chaque tour de l'interlocuteur sélectionné, et annote chaque stratégie en les délimitant par des balises. Un tour de parole est annoté dans son entièreté, avec une ou plusieurs stratégies de dialogue consécutives. Cette annotation permet de diviser un tour de parole en une séquence de segments consécutifs et sans chevauchements.

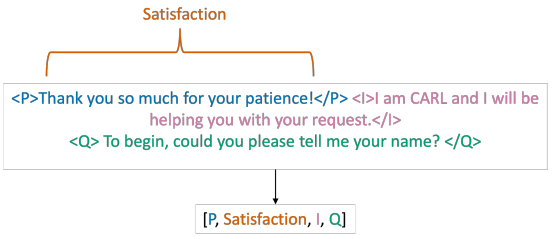


FIGURE 1 – Annotation d'un tour de parole en émotion et stratégie de dialogue.
Ici, P code les formules de politesse, I l'information et Q la question.

Au niveau de la conversation Pour l'annotation au niveau de la conversation, nous avons défini trois aspects qui permettent d'évaluer le ressenti global de la conversation sur plusieurs plans. Ces indices prennent en considération le point de vue du client et la qualité du travail de l'agent. Ils permettent également le filtrage des conversations, car nous souhaitons entraîner nos modèles sur de bons exemples de conversations avec un support agent de bonne qualité pour assurer la satisfaction du client au terme de l'interaction.

— **Satisfaction de l'utilisateur** (Satisfait - Neutre - Insatisfait) annotée par l'annotateur de la stratégie de dialogue du client, à la fin de l'annotation du fichier. Cet indicateur se base sur le comportement des clients et de leur réaction à l'interaction globale avec l'agent.

— **Qualité du support de l'agent** (Bon - Neutre - Mauvais) annotée par l'annotateur de la stratégie de dialogue de l'agent. Cet indicateur mesure le comportement de l'agent, et la justesse de ses réponses et interactions avec le client selon la situation de ce dernier.

— **Statut de résolution du problème** (Résolu - Incertain - Non résolu) annoté par l'annotateur de la stratégie de dialogue du client. Ce dernier indicateur signale l'état final de la demande du client, si le motif d'appel a été identifié et résolu par l'agent.

3.2.3 Réalisation de l'annotation

Phase de Calibrage

— **Calibrage** L'objectif de cette phase est d'initier les annotateurs à la tâche d'annotation. Les cinq annotateurs évaluent chacun l'échantillon sélectionné pour les trois tâches : émotions, stratégies de dialogue et indices de satisfaction globale. Ces fichiers ont été échantillonnés aléatoirement, tout en essayant de garder une bonne distribution des longueurs des conversations.

— **Mesure d'accord** Nous avons calculé l'alpha de Krippendorff pour évaluer les niveaux d'accord et sommes arrivés à un alpha de 0,674. La raison pour laquelle nous avons choisi l'alpha de Krippendorff, plutôt que le Kappa de Cohen par exemple, est car il s'agit d'une métrique qui prend en compte l'annotation multi-label et les annotateurs multiples. Cependant, elle ne tient pas compte de l'ordre des labels, uniquement du compte global de chaque annotation par tour.

Annotation complète Après avoir confirmé que les mesures d'accord étaient supérieures à notre seuil d'acceptation, nous avons lancé le processus d'annotation sur les échantillons restants.

3.3 Analyse du Corpus Annoté

Le corpus Le jeu de données annoté est composé de 67 conversations, ce qui représente 3051 tours de parole et 109 841 tokens. L'annotation émotionnelle a été faite avec 20 labels émotionnels secondaires, mais pour des raisons de lisibilité et de clarté, nous avons décidé de mener cette analyse en reprenant les émotions primaires correspondantes (par exemple *agacement* est une émotion secondaire de *peur*). Notre liste de labels est donc composée de 6 labels émotions et de 15 stratégies de dialogue, présentés en Table 3. La longueur moyenne des tours de parole est de 29 tokens. Ces tours de parole sont divisés en segments délimités par ces stratégies de dialogue consécutives, et leur longueur moyenne est de 17 tokens. Le nombre de labels socio-émotionnels l_i moyen par tour de parole est de 2.5 avec un l_i maximum de 23 labels socio-émotionnels, et un l_i minimum de 1 (chaque tour porte au moins un label de stratégie de dialogue). Nous donnons plus de détails sur les différences entre agent et client dans la Table 1. Nous remarquons également que les agents ont tendance à produire des tours de paroles plus longs et riches en stratégies. Une médiane de 2 labels par tour de parole valide notre approche multi-label pour la planification du prochain tour de parole. En effet, il y a en général plusieurs stratégies qui se succèdent, et reproduire le comportement d'un agent implique la prise en considération de ces multiples éléments successifs au sein d'un même tour de parole.

Les 3051 tours de paroles sont donc segmentés en 6852 segments dénotés par les stratégies de dialogue, auxquels se superposent 785 labels émotionnels. Cette différence entre le nombre d'émotions et de stratégies de dialogue s'explique par le protocole : plusieurs stratégies de dialogues sont données par tour de parole et tous les tours de parole sont entièrement annotés en stratégie de dialogue (il n'existe aucun tour de parole neutre sans aucune annotation de stratégie), alors que tous les tours ne sont pas porteurs d'émotion. Lorsqu'ils le sont c'est en général une unique émotion qui est représentée. Nous notons que la majorité des conversations propose un service qui convient au client et que dans 63 conversations, l'agent apporte un bon niveau de support.

Distribution des labels Nous avons ensuite étudié la distribution des labels annotés. La liste complète de ces labels est présentée dans la Table 3. Pour les émotions, la satisfaction est l'émotion dominante dans le discours des agents alors que pour les clients, c'est l'agacement. Certaines émotions sont très peu observées, comme la rage ou la détresse, ce qui est attendu dans le contexte des interactions que nous avons choisi. Pour ce qui est des stratégies de dialogue, la stratégie *information* est la plus représentée chez les deux interlocuteurs. En effet, celle-ci est utilisée pour apporter une information, répondre à une question ou indiquer un choix. Au niveau de la fréquence d'utilisation des stratégies,

	Agent	Client
Minimum	1	1
Maximum	16	23
Moyenne	2.7	2.3
Médiane	2	2

TABLE 1 – Statistiques liées à l_i, le nombre de labels socio-émotionnels associés au tour de parole i.

Stratégie	Total	Agent	Client
Émotions	785	382	403
Stratégies de Dialogue	6852	3582	3270
Total	7637	3964	3673
Indice	**Satisfait**	**Neutre**	**Insatisfait**
Satisfaction Utilisateur	45	15	7
	Bon	**Neutre**	**Mauvais**
Qualité Support Agent	64	3	0
	Résolu	**Incertain**	**Non Résolu**
Résolution du Problème	36	28	3

TABLE 2 – Nombre de labels annotés au niveau du tour de parole et au niveau de la conversation.

(a) Émotions

Emotion	Total	Agent	Client
Colère (*impatience, agacement, rage*)	203	42	161
Surprise	32	16	16
Peur (*stress, inquiétude, confusion*)	115	26	89
Joie (*soulagement, satisfaction*)	381	271	110
Tristesse (*déception, résignation*)	45	19	26
Dégoût (*mépris, moquerie*)	9	5	4

(b) Stratégies de Dialogue

Stratégie (Code)	Total	Agent	Client
Accord (A)	20	7	13
Aggressivité (AGR)	11	0	11
Back-Channeling (BC)	645	286	359
Correction (C)	66	18	48
Désaccord (D)	8	1	7
Encouragement (E)	52	43	9
Hors-Sujet (HS)	235	92	143
Information (I)	3023	1368	1655
Politesse (P)	721	367	354
Proposition de Suggestions (PS)	86	62	24
Question (Q)	1213	906	307
Reformulation (R)	439	328	111
Sympathie (S)	49	42	7
Self-Disclosure (SD)	240	55	185
Autre (U)	44	7	37

TABLE 3 – Compte de labels par émotion et stratégie de dialogue.

nous remarquons que pour le client ce sont celles de *question* et d'*information* qui dominent. Cela correspond au script général que les agents suivent pour qualifier puis traiter la demande client en réponse à une sollicitation client. Pour les clients, c'est la stratégie d'*information* qui est majoritaire, et constitue souvent une réponse aux questions de l'agent qui les guide à travers le processus. En effet, les 906 questions de l'agent sont réparties sur 773 tours de paroles différents (un unique tour de parole peut porter plusieurs questions), dont 619 sont suivis par une information donnée par le client.

Patterns dans les Labels Nous avons observé des patterns récurrents dans la succession des labels dans un tour de parole. Nous nous intéressons dans un premier temmps aux successions de 2 et 3 labels répétés dans le même ordre dans les énoncés du corpus. Les motifs les plus communs sont présentés en Table 4.

(a) k = 2

Pattern	Count
Information, Question	222
Information, Information	178
Question, Information	168
Back-Channeling, Information	133
Back-Channeling, Back-Channeling	127
Information, Joie	126
Reformulation, Information	100
Information, Politeness	98
Politesse, Information	85
Information, Back-Channeling	85

(b) k = 3

Pattern	Count
Information, Question, Information	92
Question, Information, Question	51
Back-channeling, Back-channeling, Back-channeling	34
Back-channeling, Back-channeling, Information	31
Information, Joie, Question	24

TABLE 4 – Motifs les plus communs pour 2 labels et 3 labels successifs dans le même tour de parole.

La plupart de ces motifs inclue des stratégies de dialogue, telles qu'*Information* ou *Back-Channeling*. Cependant, nous remarquons que les émotions sont aussi représentées, notamment par le label *Joie*. Les labels ont été annoté séparément, ce qui signifie qu'une succession de deux questions sera annoté 'Question, Question', ce qui explique pourquoi les labels peuvent se répété au sein d'un même tour (par exemple, *Back-Channeling* est une stratégie répétitive par nature, car cela implique d'interjecter pendant une prise de parole d'un interlocuteur pour le montrer qu'on prête attention). Ces patterns

nous donnent une idée de comment la plupart des réponses sont planifiées et exécutées de manière consciente ou non par les agents humains.

4 Discussion

Nous nous sommes posé la question du format des données d'entrée et de sortie de cette tâche de prédiction, particulièrement au niveau de l'unité d'annotation considérée. Dans cet article, nous utilisons le tour de parole, mais notre protocole d'annotation permet de considérer une autre unité : le segment. Chaque tour de parole est annoté en une ou plusieurs stratégies de dialogues consécutives, qui délimitent des segments. Cela impliquerait donc la définition d'un ensemble de segments c_i^j associés au tour de parole c_i, et que, pour chaque segment j, il existe une liste de labels $y_i^{j,k}$ de labels socio-émotionnels associés à ce segment. La tâche de génération serait donc reportée à l'échelle du segment, ce qui ajoute la difficulté supplémentaire de fusionner ces différents segments pour former l'unique énoncé final c_{t+1}. Nous pensons que la planification de la génération du prochain tour de parole dans son ensemble est plus pertinente, et nous avons donc choisi de conserver l'échelle du tour de parole, sans segmentation.
Il est important de noter que l'annotation comprend des biais liés aux expériences personnelles aussi bien que culturelles des annotateurs, qui peuvent influencer leurs perception des émotions et des interactions. Nous avons mis au point la phase de calibrage pour pouvoir privilégier au mieux la communication, l'extraction et l'analyse des possibles divergences, biais et incompréhensions. Cette première annotation nous a permis, au-delà de sortir les mesures d'accord, d'établir des règles et des consignes plus précises d'annotation, agrémentées d'exemples issus des données. De plus, nous avons choisi une fine granularité des annotations, pour améliorer l'explicabilité du contenu social généré. Enfin, nous n'avons pas encore ajouté de profils persona, mais nous considérons explorer cette piste à l'avenir. Nous souhaitons explorer certaines pistes, comme identifier les différents agents humains dans nos données et baser des personas à partir des données associées à ces agents. Nous pouvons également concevoir des fiches de personas fictives élaborées en collaboration avec les clients. Il serait ainsi intéressant d'étudier les différences dans les stratégies de dialogues et labels émotionnels observés selon les personas implémentées.

5 Conclusion et prochains travaux

Dans cet article, nous présentons une nouvelle approche pour la tâche de génération, en ajoutant une étape de prédiction d'une séquence de labels d'émotion et de stratégie de dialogue attendus dans le tour suivant. Nous détaillons également le protocole d'annotation associé que nous avons implémenté sur un corpus orienté-tâche. À l'issu de l'analyse de ce jeu de données annoté, la richesse du contenu émotionnel et dialogique nous conforte dans notre choix de données humain-humain, spontanées et issues d'un contexte orienté-tâche réel. L'analyse montre également qu'un tour de parole est souvent porteur de plus d'un seul label socio-émotionnel (médiane de 2 labels par tour) et l'importance des liens entre ces différents labels. Cela justifie notre approche de prédiction d'une séquence multi-label pour planifier et générer le prochain tour de parole du système conversationnel.
Nos futurs travaux utiliseront ce jeu de données annoté pour développer un système conversationnel génératif orienté-tâche capable de s'adapter de manière dynamique aux évolutions du contexte émotionnel et social de la conversation.

Références

ADIKARI A., DE SILVA D., MORALIYAGE H., ALAHAKOON D., WONG J., GANCARZ M., CHACKOCHAN S., PARK B., HEO R. & LEUNG Y. (2022). Empathic conversational agents for real-time monitoring and co-facilitation of patient-centered healthcare. *Future Generation Computer Systems*, **126**, 318–329. DOI : https ://doi.org/10.1016/j.future.2021.08.015.

BUNT H. (2006). Dimensions in dialogue act annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy : European Language Resources Association (ELRA).

CLAVEL C., ADDA G., CAILLIAU F., GARNIER-RIZET M., CAVET A., CHAPUIS G., COURCINOUS S., DANESI C., DAQUO A.-L., DELDOSSI M., GUILLEMIN-LANNE S., SEIZOU M. & SUIGNARD P. (2013). Spontaneous speech and opinion detection : mining call-centre transcripts. *Language Resources and Evaluation*, **47**(4), 1089–1125.

CLAVEL C. & CALLEJAS Z. (2016). Sentiment analysis : From opinion mining to human-agent interaction. *IEEE Transactions on Affective Computing*, **7**(1), 74–93. DOI : 10.1109/TAFFC.2015.2444846.

CLAVEL C., LABEAU M. & CASSELL J. (2022). Socio-conversational systems : Three challenges at the crossroads of fields. *Frontiers in Robotics and AI*, **9**. DOI : 10.3389/frobt.2022.937825.

CUFF B., BROWN S., TAYLOR L. & HOWAT D. (2016). Empathy : A review of the concept. *Emotion Review*, **8**, 144–153. DOI : 10.1177/1754073914558466.

EKMAN P. (1999). Basic emotions. *Handbook of cognition and emotion*, **98**(45-60), 16.

FENG S., LUBIS N., GEISHAUSER C., LIN H.-C., HECK M., VAN NIEKERK C. & GAŠIĆ M. (2021). Emowoz : A large-scale corpus and labelling scheme for emotion recognition in task-oriented dialogue systems. DOI : 10.48550/ARXIV.2109.04919.

FUNG P., BERTERO D., XU P., PARK J. H., WU C.-S. & MADOTTO A. (2018). Empathetic dialog systems. In *The international conference on language resources and evaluation. European Language Resources Association*.

GALESCU L., TENG C. M., ALLEN J. & PERERA I. (2018). Cogent : A generic dialogue system shell based on a collaborative problem solving model. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, p. 400–409.

GNEWUCH U., MORANA S. & MAEDCHE A. (2017). Towards designing cooperative and social conversational agents for customer service. In *ICIS*.

GUIBON G., LABEAU M., FLAMEIN H., LEFEUVRE L. & CLAVEL C. (2021). Few-shot emotion recognition in conversation with sequential prototypical networks. *CoRR*, **abs/2109.09366**.

HARDY A., PARANJAPE A. & MANNING C. D. (2021). Effective social chatbot strategies for increasing user initiative. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 99–110.

HARILAL N., SHAH R., SHARMA S. & BHUTANI V. (2020). Caro : An empathetic health conversational chatbot for people with major depression. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, CoDS COMAD 2020, p. 349–350, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3371158.3371220.

HOSSEINI M. & CARAGEA C. (2021). It takes two to empathize : One to seek and one to provide. In *Proceedings of the AAAI Conference on Artificial Intelligence. To appear*.

KIM H., KOH D. Y., LEE G., PARK J.-M. & LIM Y.-K. (2019). Designing personalities of conversational agents. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, p. 1–6, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3290607.3312887.

KUMAR H., AGARWAL A. & JOSHI S. (2018). Dialogue-act-driven conversation model : An experimental study. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 1246–1256, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

LI J., GALLEY M., BROCKETT C., SPITHOURAKIS G. P., GAO J. & DOLAN B. (2016). A persona-based neural conversation model. *arXiv preprint arXiv :1603.06155*.

LI Y., SU H., SHEN X., LI W., CAO Z. & NIU S. (2017). Dailydialog : A manually labelled multi-turn dialogue dataset. DOI : 10.48550/ARXIV.1710.03957.

LIU S., ZHENG C., DEMASI O., SABOUR S., LI Y., YU Z., JIANG Y. & HUANG M. (2021). Towards emotional support dialog systems. *ArXiv*, **abs/2106.01144**.

LU X., TIAN Y., ZHAO Y. & QIN B. (2021). Retrieve, discriminate and rewrite : A simple and effective framework for obtaining affective response in retrieval-based chatbots. In *Findings of the Association for Computational Linguistics : EMNLP 2021*, p. 1956–1969.

MASLOWSKI I., LAGARDE D. & CLAVEL C. (2017). In-the-wild chatbot corpus : from opinion analysis to interaction problem detection. In *ICNLSSP 2017*, p. 115–120, Casablanca, Morocco : ISGA, Institut Supérieur d'inGénierie et des Affaires. HAL : hal-02288505.

MAZARÉ P.-E., HUMEAU S., RAISON M. & BORDES A. (2018). Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2775–2779, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1298.

PLUTCHIK R. (1984). Emotions : a general psychoevolutionary theory.

PRADHAN A. & LAZAR A. (2021). Hey google, do you have a personality ? designing personality and personas for conversational agents. In *Proceedings of the 3rd Conference on Conversational User Interfaces*, CUI '21, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3469595.3469607.

RAM A., PRASAD R., KHATRI C., VENKATESH A., GABRIEL R., LIU Q., NUNN J., HEDAYATNIA B., CHENG M., NAGAR A., KING E., BLAND K., WARTICK A., PAN Y., SONG H., JAYADEVAN S., HWANG G. & PETTIGRUE A. (2018). Conversational ai : The science behind the alexa prize. DOI : 10.48550/ARXIV.1801.03604.

RASHKIN H., SMITH E. M., LI M. & BOUREAU Y.-L. (2018). Towards empathetic open-domain conversation models : A new benchmark and dataset. *arXiv preprint arXiv :1811.00207*.

SANTOS TEIXEIRA M. & DRAGONI M. (2022). A review of plan-based approaches for dialogue management. *Cognitive Computation*, p. 1–20.

SHUSTER K., XU J., KOMEILI M., JU D., SMITH E. M., ROLLER S., UNG M., CHEN M., ARORA K., LANE J., BEHROOZ M., NGAN W., POFF S., GOYAL N., SZLAM A., BOUREAU Y.-L., KAMBADUR M. & WESTON J. (2022). Blenderbot 3 : a deployed conversational agent that continually learns to responsibly engage. DOI : 10.48550/ARXIV.2208.03188.

SKERRY A. E., SAXE R., SKERRY A. E. & SAXE R. (2015). Neural representations of emotion are organized around abstract event features. *Curr. Biol.*

THOPPILAN R., DE FREITAS D., HALL J., SHAZEER N., KULSHRESHTHA A., CHENG H.-T., JIN A., BOS T., BAKER L., DU Y., LI Y., LEE H., ZHENG H. S., GHAFOURI A., MENEGALI M., HUANG Y., KRIKUN M., LEPIKHIN D., QIN J., CHEN D., XU Y., CHEN Z., ROBERTS A., BOSMA M., ZHAO V., ZHOU Y., CHANG C.-C., KRIVOKON I., RUSCH W., PICKETT M., SRINIVASAN P., MAN L., MEIER-HELLSTERN K., MORRIS M. R., DOSHI T., SANTOS R. D., DUKE T., SORAKER J., ZEVENBERGEN B., PRABHAKARAN V., DIAZ M., HUTCHINSON B., OLSON K., MOLINA A., HOFFMAN-JOHN E., LEE J., AROYO L., RAJAKUMAR R., BUTRYNA A., LAMM M., KUZMINA V., FENTON J., COHEN A., BERNSTEIN R., KURZWEIL R., AGUERA-ARCAS B., CUI C., CROAK M., CHI E. & LE Q. (2022). Lamda : Language models for dialog applications. DOI : 10.48550/ARXIV.2201.08239.

VANEL. L., YACOUBI. A. & CLAVEL. C. (2023). A survey of socio-emotional strategies for generation-based conversational agents. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence - Volume 3 : ICAART,*, p. 185–192 : INSTICC SciTePress. DOI : 10.5220/0011632400003393.

WANG Y.-H., HSU J.-H., WU C.-H. & YANG T.-H. (2021). Transformer-based empathetic response generation using dialogue situation and advanced-level definition of empathy. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, p. 1–5. DOI : 10.1109/ISCSLP49672.2021.9362067.

WELIVITA A. & PU P. (2020). A taxonomy of empathetic response intents in human social conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4886–4899, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.429.

WELIVITA A., XIE Y. & PU P. (2021). A large-scale dataset for empathetic response generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 1251–1264.

ZANDIE R. & MAHOOR M. H. (2020). Emptransfo : A multi-head transformer architecture for creating empathetic dialog systems. *CoRR*, **abs/2003.02958**.

ZHANG S., DINAN E., URBANEK J., SZLAM A., KIELA D. & WESTON J. (2018). Personalizing dialogue agents : I have a dog, do you have pets too ? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2204–2213, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1205.

ZHANG Y., SUN S., GALLEY M., CHEN Y.-C., BROCKETT C., GAO X., GAO J., LIU J. & DOLAN B. (2019). Dialogpt : Large-scale generative pre-training for conversational response generation. DOI : 10.48550/ARXIV.1911.00536.

ZHONG P., ZHANG C., WANG H., LIU Y. & MIAO C. (2020). Towards persona-based empathetic conversational models. *arXiv preprint arXiv :2004.12316*.

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Exploring Data-Centric Strategies for French Patent Classification: A Baseline and Comparisons

You Zuo[1,2] Houda Mouzoun[3] Samir Ghamri Doudane[3]
Kim Gerdes[1,4] Benoît Sagot[2]
(1) Qatent, Paris, France
(2) Inria, Paris, France
(3) Institut national de la propriété industrielle, Courbevoie, France
(4) Laboratoire Interdisciplinaire des Sciences du Numérique, Université Paris-Saclay, Orsay, France
firstname.lastname@inria.fr, hmouzoun@inpi.fr, sghamridoudane@inpi.fr, gerdes@lisn.fr

ABSTRACT

This paper proposes a novel approach to French patent classification leveraging data-centric strategies. We compare different approaches for the two deepest levels of the IPC hierarchy : the IPC group and subgroups. Our experiments show that while simple ensemble strategies work for shallower levels, deeper levels require more sophisticated techniques such as data augmentation, clustering, and negative sampling. Our research highlights the importance of language-specific features and data-centric strategies for accurate and reliable French patent classification. It provides valuable insights and solutions for researchers and practitioners in the field of patent classification, advancing research in French patent classification.

RÉSUMÉ

Exploration des stratégies centrées sur les données pour la classification des brevets français : Une base de référence et des comparaisons

Cet article propose une nouvelle approche de classification des brevets français qui s'appuie sur des stratégies centrées sur les données. Nous comparons différentes approches pour les deux niveaux les plus profonds de la hiérarchie IPC : le groupe IPC et les sous-groupes. Nos expériences montrent que les stratégies d'ensemble simples fonctionnent pour les niveaux peu profonds, mais que les niveaux profonds nécessitent des techniques plus sophistiquées telles que l'augmentation de données, le regroupement et l'échantillonnage négatif. Notre recherche met en évidence l'importance des caractéristiques spécifiques à la langue et des stratégies centrées sur les données pour une classification précise et fiable des brevets français. Elle fournit des informations et des solutions précieuses pour les chercheurs et les praticiens dans le domaine de la classification des brevets, en faisant progresser la recherche en classification des brevets en français.

KEYWORDS : Patent Classification, Extreme Multi-label Text Classification, Deep Learning.

MOTS-CLÉS: Classification de Brevets d'Invention, Classification de Textes Multi-labels Extrême, Apprentissage Profond.

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 349–365.

1 Introduction

A patent is a legal document that grants its holder the exclusive right to prevent others from making, using, selling, offering for sale, or importing the patented invention within a certain jurisdiction for a specific period of time. Patent databases are valuable sources of information that reflect global innovations and technological developments. The number of patent applications has increased significantly over the past two decades, which can be attributed to various factors such as the increasing importance of technology in society, the role of patents in business valuation, and the globalization of commerce. Due to the large volume of patents and patent documents, patent analysis and management have become complex and time-consuming. Additionally, patents are granted within a specific domain, and patent classification is critical for patent scope and patent law. For the reasons mentioned above, automated patent classification systems have become essential for patent professionals to manage large collections of patents.

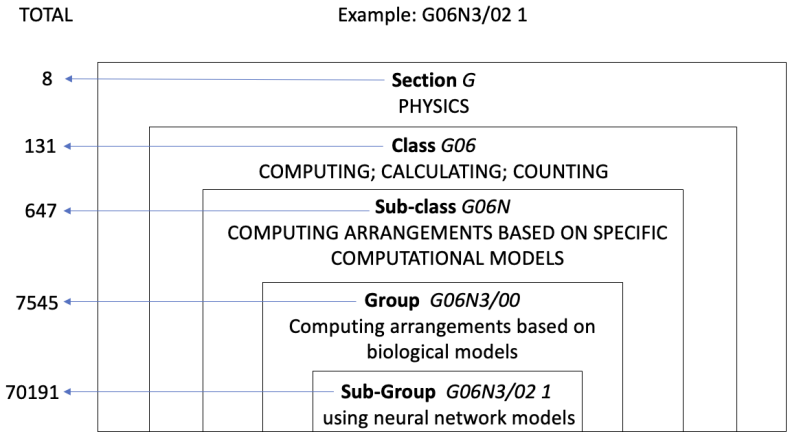


FIGURE 1 – An example of International Patent Classification (IPC) code G06N3/02 1. The IPC scheme is a hierarchical taxonomy with five levels.

The International Patent Classification (IPC) [1] is a widely used standard taxonomy for patent classification. It comprises a hierarchy of five levels, including section, class, subclass, group, and subgroup. The finest level, the subgroup, consists of over 70,000 leaf nodes. An example of an IPC code is shown in Figure 1. When submitting a patent application, the relevant technical fields are indicated by assigning one or multiple IPC codes at the subgroup level. This helps determine the workload of examiners responsible for assessing the application.

However, the IPC undergoes regular minor updates and occasionally major restructuring. Developing an automatic patent classification system is thus challenging due to the constantly evolving technological language and patent syntax, requiring ongoing efforts to address this moving target. In addition, the classification of patent text is a widely studied problem, particularly in languages with large patent markets, such as English, Chinese, and Japanese. However, due to variations in technical fields across different countries and markets, it can be challenging to directly transfer models or data trained on English patents to other languages. This problem is particularly challenging in the case of French patents, in particular presents unique challenges that require a sizable annotated corpus and experimentation with various classification models. Despite the critical importance of French patents,

1. https://www.wipo.int/classifications/ipc/

few studies, such as (Zuo *et al.*, 2022), has focused on patent classification in French, and even those have only reported results on IPC subclass or group levels.

To address this gap, this paper proposes a novel approach to French patent classification that builds upon different strategies aimed at addressing several unique challenges, including input length limitation, label imbalance, and data sparsity. Our approach compares and evaluates different strategies for French patent classification at the two deepest levels of the IPC hierarchy : the IPC group and subgroups. To the best of our knowledge, this is the first comprehensive work specifically focused on patent classification at such deep levels for the French language.

2 Background

A patent is a well-structured document that typically includes several sections, such as a title, abstract, background, brief summary of the invention, detailed description, one or more claims, drawings, and classification information. The International Patent Classification (IPC) is a widely used system for uniformly classifying the content of patents, with over 100 countries currently employing it. The IPC scheme, which was established by the World Intellectual Property Organization (WIPO) [2] in 1971, is hierarchical and serves as the preferred classification system for French patent classification. The IPC system consists of eight general categories of patents, represented by the section level of IPC. These categories are hierarchical and represent the broadest possible areas of technology, providing a starting point for patent classification. Table 1 illustrates the eight most general categories of patents.

Section	Title
A	HUMAN NECESSITIES
B	PERFORMING OPERATIONS ; TRANSPORTING
C	CHEMISTRY ; METALLURGY
D	TEXTILES ; PAPER
E	FIXED CONSTRUCTIONS
F	MECHANICAL ENGINEERING ; LIGHTING ; HEATING ; WEAPONS ; BLASTING
G	PHYSICS
H	ELECTRICITY

TABLE 1 – The eight IPC section categories.

In practice, patent offices assign classification codes to patent applications to accurately describe the subject matter of the invention. They typically assign the most specific level of the International Patent Classification (IPC) to classify the documents in their database. By assigning a specific subgroup level of IPC, broader groups and higher levels of the IPC hierarchy that encompass it can be determined, as the IPC hierarchy is organized in a tree-like structure. This avoids any double classification of "parent" and "child" classes in the IPC hierarchy.

In this work, we focus on exploring the effectiveness of classification methods on groups and subgroups separately.

2. https://www.wipo.int

3 Related Work

Prior research on automated patent classification systems has typically relied on traditional algorithms and feature extraction methods (Verberne & D'hondt, 2011; Yun & Geum, 2020; Wu *et al.*, 2010; Cai & Hofmann, 2007; Qiu *et al.*, 2011). However, designing hand-crafted features can be time-consuming and lead to efficiency problems, making them difficult to apply to large patent collections.

More recently, researchers have turned to deep learning techniques to leverage large-scale training data and generalize well to unseen data. For example, (Grawe *et al.*, 2017) used LSTM on 50 IPC subgroups, while DeepPatent (Li *et al.*, 2018) employed a CNN with skip-gram word embeddings. (Risch & Krestel, 2019) trained fastText word embeddings on full-text patents and then used GRU models on top of these embeddings. Pre-trained language models, such as ULMFiT (Hepburn, 2018) and BERT (Lee & Hsiang, 2019), have also been fine-tuned for this specific task. Comparing the performance of different pre-trained models for multi-label patent classification, XLNet (Yang *et al.*, 2020) was found to outperform other models (Roudsari *et al.*, 2021). Moreover, (Zhang *et al.*, 2022) reduced the uncertainty of classification results through the fusion of multiple patent views. Ensemble techniques have also been explored in patent classification. For instance, (Kamateri *et al.*, 2022) found that an ensemble of classifiers with separate inputs for title-abstract, claims, and description achieved the best performance.

Some researchers have explored mapping patent texts to International Patent Classification (IPC) codes using KNN classification and hybrid methods that combine neural feature encoders with KNN (Cai *et al.*, 2010; Bekamiri *et al.*, 2021, 2022). Another approach is to use a Wide and Deep (WnD) network (Cheng *et al.*, 2016) to combine string-level similarity and semantic embeddings of patent text (Niu & Cai, 2019).

The previous research on patent classification has primarily focused on flat classification, where the classification problem is considered at one specific shallow level of the hierarchy, such as the class or subclass level of the IPC. Only a few studies have addressed the more detailed classification of patent data at the group and subgroup levels. For instance, (Chen & Chang, 2012) proposes a three-phase categorization algorithm using SVM classifiers, (Risch *et al.*, 2020) formulates the hierarchical classification problem as a sequence generation task, and (Zuo *et al.*, 2022) formulates the patent classification problem at deeper levels as an extreme multi-label text classification (XMTC) task.

Despite these efforts, many of these works still rely on feature extraction and model architecture improvements, which only partially address the challenges of accurate patent classification. In contrast, our work focuses on enhancing the quality of the training set and its input format for automatic patent classification, leveraging the same dataset proposed in (Zuo *et al.*, 2022), to improve model performance specifically at the most detailed levels of the IPC hierarchy.

4 Methodology

The task can be framed as follows : given a patent document x, it is assigned with one or more IPC codes $l \in \mathcal{L} = \{l_1, l_2, \ldots, l_L\}$, where L represents the total number of predefined IPC codes at a specific level. Our training set $\mathcal{D} = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \{0,1\}^L, i = 1, 2, \ldots, N\}$ comprises instances x_i and their corresponding labels y_i, with the objective of learning a scoring function f. This function maps an input x_i and a label l to a score $f(x_i, l) \in \mathbb{R}$ and is optimized to maximize

the score when $y_{i,l} = 1$ (i.e., the label l is relevant to the instance x_i) and minimize the score when $y_{i,l} = 0$ (i.e., the label l is irrelevant to the instance x_i).

4.1 Baseline Model

We selected LightXML (Jiang *et al.*, 2021) as our baseline model because it has shown the best performance on INPI-CLS according to (Zuo *et al.*, 2022). LightXML is a transformer-based model specifically designed for multi-label text classification in English. It employs multiple pre-trained language models, such as BERT (Devlin *et al.*, 2018), Roberta (Liu *et al.*, 2019), and XLNet (Yang *et al.*, 2020), and concatenates the representations of the special token [CLS] in the last five hidden states to create a text representation. To handle negative sampling during training, LightXML utilizes a label-recalling network to dynamically sample negative samples during training, followed by a label-ranking network to separate positive from negative labels. We obtained the codes we used from the online extreme classification repository (Bhatia *et al.*, 2016) [3].

Since our corpus is in French, we also used three BERT-like language models specifically designed for French (Martin *et al.*, 2020) or multilingual contexts (Pires *et al.*, 2019; Conneau *et al.*, 2019). The final result of LightXML is the ensemble of the three classifiers. In Table 2, we provide a preliminary comparison of the selected pre-trained models.

Checkpoint	camembert-base (Martin *et al.*, 2020)	bert-base-multilingual-cased (Pires *et al.*, 2019)	xlm-roberta-base (Conneau *et al.*, 2019)
Pre-training Dataset	OSCAR(Suárez *et al.*, 2019)	Wikipedia	cleaned CommonCrawl
Pre-training Tasks	MLM	MLM + NSP	MLM
# Languages	1	104	100
# Parameters	110M	110M	125M
Tokenization	SentencePiece	WordPiece	SentencePiece

TABLE 2 – Overview of pre-trained models we use for LightXML for the classification of French patents.

The methods presented in the remainder of this paper aim to enhance the performance of the baseline approach. We conduct a comprehensive ablation study to offer additional insights for future research on the patent classification problem.

4.2 Weighted Sum Ensemble

An ensemble is a collection of models designed to improve the performance of individual base models by combining their predictions. In this study, we employed the weighted sum ensemble method, which assigns a weight to each base model based on its performance on a validation set. We evaluated the performance of each base model using the $precision@k(k = 1)$ score, which measures the proportion of correct predictions among the top-ranked k categories suggested by the model. This method assumes that some models in the ensemble are more effective than others.

To build our ensemble, we first trained multiple base models using different architectures or training strategies. Next, we evaluated the performance of each model using a separate validation set and

3. http://manikvarma.org/downloads/XC/XMLRepository.html

selected the top-performing ones. Finally, we combined the predictions of the selected models by taking a weighted sum of their output probabilities, where the weight of each model is proportional to its precision@1 score.

4.3 Information Extraction

Due to the maximum input length restrictions of BERT-like models (512 tokens), we explored different approaches to extract the most critical information from patent content.

TextRank (Mihalcea & Tarau, 2004). TextRank is a graph-based ranking model that identifies the most relevant sentences and keywords in text. We considered using TextRank because we observed that using a patent description as input yielded good results. However, the lengthy description text posed a challenge since only the first few hundred words could be entered. To address this issue, we used TextRank [4] to extract the most critical sentences from the patent description as input.

SAO (Subject-Action-Object) Extraction. The SAO structure is a widely used approach in patent analysis for representing technology concepts in a subject-action-object format (Choi *et al.*, 2010; Radauer & Walter, 2010). This structure can be extracted through grammatical processing of patent text and enables a more systematic understanding of the central functional properties of the patent application. For example, consider the sentence, "The super-capacitor electrode further comprising a silane coupling agent." In this sentence, the subject is "super-capacitor electrode," the action is "comprise," and the object is "silane coupling agent." By analyzing many SAO structures in a patent's text, we can extract the underlying functional relationships, identify key features of the invention, and reformulate it as input for classifier models.

To make use of the extracted SAO structures, we typically reformulate them as simple sentences for easier interpretation. For example, the SAO triplet extracted from the above sentence can be reformulated as "The super-capacitors comprise a silane coupling agent." This method [5] has been applied to the claims section of a patent, which is in a strict syntactic format and hence more amenable to rule-based extraction. The extracted SAO structures can then be used as input for our classifier models to improve their performance.

4.4 Vocabulary Enlargement

Another limitation of using pre-trained language models for patent classification is the presence of a large number of scientific and technological terminologies that are seldom encountered in the pre-training corpus, leading to suboptimal model performance. To address this challenge, we propose incorporating external vocabularies from other models. Specifically, we leverage features from sparse classifiers, such as logistic regression with TF-IDF, to extract its 100 most important terms and add them to the vocabulary of the neural encoder. Examples of lemmatized terms can be found in Appendix A.

4. https://summanlp.github.io/textrank/
5. Codes used for SAO extraction : https://github.com/ZoeYou/SAO-extraction

4.5 Sampling Strategies

Dynamic Negative Sampling. LightXML(Jiang *et al.*, 2021) offers a dynamic negative sampling approach that incorporates generative cooperation networks to recall and rank labels from recalled label clusters and dynamically sample negative labels during label ranking. However, for datasets with a small label space, the paper suggests that there is no need to build label clusters and the label recall and re-ranking module degenerates to a linear layer. It can be challenging to determine the appropriate size of the label space to decide whether to set aside the recall and ranking modules. In our study, we compare the classification performance of LightXML with and without the recall and ranking modules at the group and subgroup levels of the IPC.

Oversampling. Class imbalance is a common problem in patent classification datasets that can lead to biased models and poor performance on underrepresented classes. Oversampling is a technique that can address class imbalance by increasing the number of samples in the minority class, thereby providing a better representation of rare concepts. Different oversampling techniques are available for patent classification, such as weighted oversampling and SMOTE (Chawla *et al.*, 2002). In our study, we use the weighted oversampling strategy, where the weight of each label is represented by its inverse frequency in the dataset. We leave the exploration of other oversampling methods for future studies.

4.6 Data Augmentation

The scarcity of training data is a significant challenge for accurate French patent classification. To overcome this obstacle, we explore the possibility of leveraging annotated data from external sources. Although multilingual patent datasets, such as MAREC/IREC (Piroi, 2021) and (Roda *et al.*, 2009; Piroi, 2010; Piroi *et al.*, 2011), have been previously proposed, we opted not to use them because their publication dates significantly differ from our test set. Instead, we obtained a more recent dataset of annotated patent texts published between 2010 and 2019 from the European Patent Office (EPO). This dataset closely resembles our target test set in terms of label distribution and format.

We present a comprehensive analysis of the label distribution across IPC sections in various datasets in Appendix B. Table 3 provides a summary of the statistics of the EPO data [6] that we used.

Language	# Title	# Abstract	# Claims	# Description
French	1,087,313	352,410	507,998	29,539
English	1,082,679	591,045	1,099,062	981,128

TABLE 3 – Statistics of EPO in English and French.

Introduction of EPO French Data. We incorporated French patent data published by the EPO from 2010-2019 into our training set to address the issue of limited training data for French patent classification. Our test set remained the same throughout our work.

Translation of EPO English Data. Introducing French data from the EPO alone did not yield satisfactory performance in deeper levels of IPC classification. To overcome this challenge, we fine-tuned a T5 model (Raffel *et al.*, 2020) using various strategies to translate EPO English patents into French, which augmented the French training data. We utilized the parallel European patent dataset

6. EPO data extracted from EP full-text data for text analytics

EuroPat (Heafield *et al.*, 2022) to train our models. Further details of the fine-tuning experiments can be found in Appendix C.

5 Experiments and Results

5.1 Dataset Description

In this study, we employ the French Patent corpus INPI-CLS (Zuo *et al.*, 2022) as the dataset for our patent classification task. This corpus consists of patents extracted from the internal database of the French National Institute of Industrial Property (INPI) [7] and covers patent texts from 2002 to 2021, including the title, abstract, claims, and description. Each patent in the corpus has been labeled with IPC codes at all levels, from the most general sections to the more specific subgroup labels. In this study, we specifically focus on the group and subgroup levels of IPC, which have been identified as more challenging in previous research. We conducted a time-based split of the corpus to create separate training and test sets. The training set includes patents published between 2002 and 2019, while the test set is composed of patents published between 2020 and 2021.

Dataset	**N**	$\mathbf{L_6}$	$\mathbf{\bar{L}_6}$	$\mathbf{\hat{L}_6}$	$\mathbf{L_8}$	$\mathbf{\bar{L}_8}$	$\mathbf{\hat{L}_8}$
Train	268,254	6,788	2.21	39.52	48,932	2.73	5.48
Test	28,017	4,351	2.20	6.44	19,593	2.64	1.43

TABLE 4 – Basic Statistics of INPI-CLS dataset used for our experiments. L indicates the label count, $\bar{L}$ stands for the average number of IPC labels per patent document, and $\hat{L}$ represents the average number of patent documents per label. The subscripts 6 and 8 indicate the IPC code's length in characters for the IPC's group and subgroup levels, respectively.

5.2 Experimental Setup

We evaluate the performances of models with the rank-based metrics $Precision@K$ and $Recall@K$ $(k = 1, 3, 5)$. $Precision@K$ and $Recall@K$ are calculated for each test patent and then averaged over all the patent documents. Theoretically, each patent document is assigned firstly a primary IPC code, followed by an unlimited number of secondary IPC codes. However, during our evaluation, we did not take into account the order in which the predicted IPC codes are assigned. This aspect is left for future work.

To improve the training efficiency, we fix the maximum input length of each encoder to 128. For other configurations of LightXML, we fix $learning\_rate = 1e-4$ with warm-up, $batch\_size = 16$, and $number\_epoch = 3$.

5.3 Results and Discussion

We evaluated the performance of the LightXML model with baseline configurations at the IPC group and subgroup level, and present the results in Table 5 and Table 6, respectively. Our experiments

7. https://data.inpi.fr/recherche_avancee/brevets

show that the choice of dataset has a significant impact on the performance of the LightXML model for patent classification. In particular, we found that the title+abstract dataset achieved the best performance in terms of $Precision@k$ and $Recall@k$ values for IPC group classification, while the description dataset yielded the highest performance for IPC subgroup classification.

Moreover, our results indicate that the LightXML algorithm performs better at the IPC group level than at the IPC subgroup level. This is because the latter level requires more detailed information about the invention, and the subgroup classes are often imbalanced, making it more challenging to achieve high performance. However, we believe that our approach can be further improved by addressing these issues, such as by incorporating more relevant information or using more advanced modeling techniques.

Dataset	P@1	P@3	P@5	R@1	R@3	R@5
title+abstract	61.08	37.24	26.84	27.67	50.61	60.80
claims	58.60	35.61	25.83	26.54	48.39	58.50
description	58.99	36.30	26.26	26.72	49.33	59.48

TABLE 5 – Baseline performance at IPC group level.

Dataset	P@1	P@3	P@5	R@1	R@3	R@5
title+abstract	12.59	8.80	7.12	4.76	9.99	13.47
claims	15.34	10.65	8.50	5.80	12.09	16.07
description	18.52	12.54	9.87	7.01	14.23	18.67

TABLE 6 – Baseline performance at IPC subgroup level.

Furthermore, in our experiments comparing the performance of classifiers based on different encoders (as shown in Figure 2), we consistently found that `mbert` outperformed other encoders. These results suggest that `mbert` is highly effective in capturing the nuances of the French language in patent documents. One possible reason for its superior performance is that it is trained on a large and diverse corpus, which includes the Wikipedia corpus that contains a vast range of technical terms and topics relevant to patent texts.

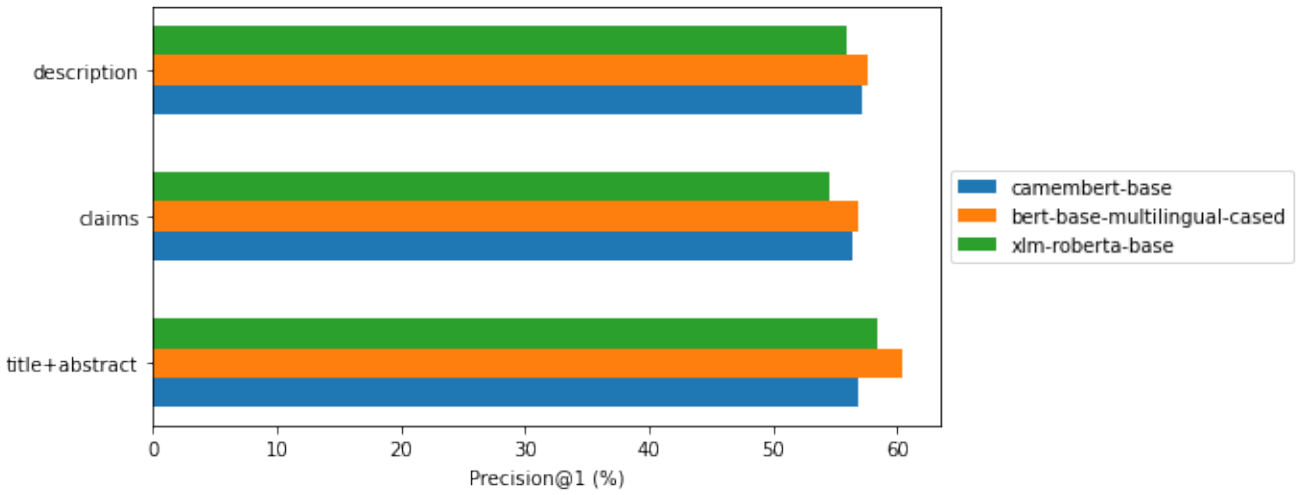


FIGURE 2 – P@1 of classifiers based on different encoders (models trained on IPC group level).

Our experiments comparing different strategies for French patent classification are summarized in Tables 7 and 8. Notably, our results indicate that the weighted sum ensemble approach was not always effective in improving classifier performance, particularly for the subgroup level when the ensemble includes a basic classifier with poor performance. We also observed that methods for extracting the most essential information from patent texts, such as TextRank and SAO extraction, led to a significant decrease in classification performance, particularly for the claims dataset. One possible reason for this is that these methods may have removed important details and nuances from the patent texts that are necessary for accurate classification. In addition, TextRank may have selected sentences from different subsections of the patent description that serve different functions, resulting in a loss of coherence among sentences.

Furthermore, we discovered that enlarging the vocabulary did not enhance the baseline performance. One possible reason for this could be that the additional vocabulary was not well-tuned during the pre-training process, and fine-tuning only the last five feature layers did not provide accurate semantic meaning for the added vocabulary.

Our experimental results for dynamic negative sampling indicate that this technique, proposed in the original LightXML paper, had varying effects on the classifiers' performance at different levels of the IPC hierarchy. Specifically, while this technique led to a decrease in performance for classifiers trained at the IPC group level, it resulted in significant performance improvements for classifiers trained at the subgroup level. These results suggest that in the context of multi-label classification, where long-tail distribution is a significant challenge, clustering techniques and effective negative sampling methods can greatly enhance classification performance.

Moreover, our experiments also showed that the oversampling technique, which involved duplicating minority samples to address class imbalance, did not always improve performance. This suggests that weighted oversampling of long-tail labels may not be an effective approach for addressing label imbalance in French patent classification.

To improve our classification performance, we also investigated the use of supplementary training data, including French data sourced from the EPO, as well as English data translated to French using neural machine translation. We focused on incorporating claims data from the EPO dataset, as it is typically longer and more complete than other sections of patents such as abstracts and descriptions. Our experiments show that the introduction of additional French data from the EPO led to a significant improvement in performance, as demonstrated in Table 8. Furthermore, our models were able to benefit from the increased volume and variety of training examples when augmenting the EPO French data with EPO English data, even when part of the latter dataset was translated from English.

Methods	**title+abstract**	**claims**	**description**
Baseline	61.08	58.60	58.99
Ensemble		63.97	
TextRank		47.11	56.35
SAO Extraction		47.57	
Vocabulary Enlargement	60.98	58.52	58.08
Dynamic Negative Sampling	54.16	48.90	52.38
Oversampling	56.86	53.62	55.47

TABLE 7 – Overall Precision@1 of proposed methods on IPC group level.

Methods	title+abstract	claims	description
Baseline	12.59	15.34	18.52
Ensemble		17.40	
Dynamic Negative Sampling	28.93	26.91	28.18
++data EPO_fr		32.43	
++data EPO_fr & EPO_en by NMT		34.06	

TABLE 8 – Overall Precision@1 of proposed methods on IPC subgroup level.

6 Conclusion

In this paper, we proposed and compared various data-centric approaches for French patent classification. Through extensive experiments, we demonstrated that an ensemble strategy can significantly improve patent classification at shallower levels, such as the IPC group level. However, for deeper levels of classification, such as the IPC subgroup level, where data scarcity and long-tail label distribution are common problems, we recommend using data augmentation techniques, clustering, and negative sampling during the training process to improve model performance.

Our research makes a valuable contribution to the development of automated patent classification systems in the French language, and we hope that our findings will inspire further research in this area. In summary, our work highlights the potential of data-centric strategies to overcome the challenges associated with patent classification and lays the groundwork for future studies in this field.

Acknowledgements

This work was funded and conducted within the framework of a cooperation agreement between the French National Industrial Property Office (INPI) and the French National Institute for Research in Computer Science and Automation (INRIA) for the development of a patent classification model. We would like to express our gratitude to the CLEPS infrastructure at Inria Paris for providing resources and support. The last author's contribution was also supported by his chair in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

References

BEKAMIRI H., HAIN D. S. & JUROWETZKI R. (2021). Hybrid model for patent classification using augmented SBERT and KNN. *CoRR*, **abs/2103.11933**.

BEKAMIRI H., HAIN D. S. & JUROWETZKI R. (2022). A survey on sentence embedding models performance for patent analysis. *arXiv preprint arXiv :2206.02690*.

BHATIA K., DAHIYA K., JAIN H., KAR P., MITTAL A., PRABHU Y. & VARMA M. (2016). The extreme classification repository : Multi-label datasets and code.

CAI L. & HOFMANN T. (2007). Exploiting known taxonomies in learning overlapping concepts. In *IJCAI*, volume 7, p. 708–713.

CAI Y. L., JI D. & CAI D. (2010). A knn research paper classification method based on shared nearest neighbor. In *NTCIR*.

CHAWLA N. V., BOWYER K. W., HALL L. O. & KEGELMEYER W. P. (2002). Smote : synthetic minority over-sampling technique. *Journal of artificial intelligence research*, **16**, 321–357.

CHEN Y.-L. & CHANG Y.-C. (2012). A three-phase method for patent classification. *Information Processing & Management*, **48**(6), 1017–1030.

CHENG H.-T., KOC L., HARMSEN J., SHAKED T., CHANDRA T., ARADHYE H., ANDERSON G., CORRADO G., CHAI W., ISPIR M. *et al.* (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, p. 7–10.

CHOI S., LIM J., YOON J. & KIM K. (2010). Patent function network analysis : A function based approach for analyzing patent information. In *19th International conference for the international association of management of technology, Cairo, Egypt, March*, p. 8–11.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv :1911.02116*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

GRAWE M. F., MARTINS C. A. & BONFANTE A. G. (2017). Automated patent classification using word embedding. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, p. 408–411 : IEEE.

HEAFIELD K., FARROW E., VAN DER LINDE J., RAMÍREZ-SÁNCHEZ G. & WIGGINS D. (2022). The europat corpus : A parallel corpus of european patent data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 732–740.

HEPBURN J. (2018). Universal language model fine-tuning for patent classification. In *Proceedings of the Australasian Language Technology Association Workshop 2018*, p. 93–96, Dunedin, New Zealand.

JEHL L. & RIEZLER S. (2018). Document-level information as side constraints for improved neural patent translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1 : Research Track)*, p. 1–12.

JIANG T., WANG D., SUN L., YANG H., ZHAO Z. & ZHUANG F. (2021). Lightxml : Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, p. 7987–7994.

KAMATERI E., STAMATIS V., DIAMANTARAS K. & SALAMPASIS M. (2022). Automated single-label patent classification using ensemble classifiers. In *2022 14th International Conference on Machine Learning and Computing (ICMLC)*, p. 324–330.

LEE J.-S. & HSIANG J. (2019). Patentbert : Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv :1906.02124*.

LI S., HU J., CUI Y. & HU J. (2018). Deeppatent : Patent classification with convolutional neural networks and word embedding. *Scientometrics*, **117**(2), 721–744. DOI : 10.1007/s11192-018-2905-5.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

MIHALCEA R. & TARAU P. (2004). Textrank : Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, p. 404–411.

NIU M. & CAI J. (2019). A label informative wide & deep classifier for patents and papers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3438–3443.

PIRES T., SCHLINGER E. & GARRETTE D. (2019). How multilingual is multilingual bert ? *arXiv preprint arXiv :1906.01502*.

PIROI F. (2010). Clef-ip 2010 : Retrieval experiments in the intellectual property domain.

PIROI F. (2021). The marec/irec data set. DOI : 10.48436/2zx6e-5pr64.

PIROI F., LUPU M., HANBURY A. & ZENZ V. (2011). Clef-ip 2011 : Retrieval in the intellectual property domain.

QIU X., HUANG X.-J., LIU Z. & ZHOU J. (2011). Hierarchical text classification with latent concepts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 598–602.

RADAUER A. & WALTER L. (2010). Elements of good practice for providers of publicly funded patent information services for smes–selected and amended results of a benchmarking exercise. *World Patent Information*, **32**(3), 237–245.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, **21**(1), 5485–5551.

RISCH J., GARDA S. & KRESTEL R. (2020). Hierarchical document classification as a sequence generation task. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, p. 147–155.

RISCH J. & KRESTEL R. (2019). Domain-specific word embeddings for patent classification. *Data Technologies and Applications*.

RODA G., TAIT J., PIROI F. & ZENZ V. (2009). Clef-ip 2009 : Retrieval experiments in the intellectual property domain. volume 1175, p. 385–409. DOI : 10.1007/978-3-642-15754-7_47.

ROUDSARI A. H., AFSHAR J., LEE W. & LEE S. (2021). Patentnet : multi-label classification of patent documents using deep learning based language understanding. *Scientometrics*.

SUÁREZ P. J. O., SAGOT B. & ROMARY L. (2019). Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)* : Leibniz-Institut für Deutsche Sprache.

VERBERNE S. & D'HONDT E. (2011). Patent classification experiments with the linguistic classification system lcs in clef-ip 2011. volume 1177.

WU C.-H., KEN Y. & HUANG T. (2010). Patent classification system using a new hybrid genetic algorithm support vector machine. *Applied Soft Computing*, **10**(4), 1164–1177. Optimisation Methods Applications in Decision-Making Processes, DOI : https ://doi.org/10.1016/j.asoc.2009.11.033.

YANG Z., DAI Z., YANG Y., CARBONELL J., SALAKHUTDINOV R. & LE Q. V. (2020). Xlnet : Generalized autoregressive pretraining for language understanding.

YUN J. & GEUM Y. (2020). Automated classification of patents : A topic modeling approach. *Computers & Industrial Engineering*, **147**, 106636.

ZHANG L., LIU W., CHEN Y. & YUE X. (2022). Reliable multi-view deep patent classification. *Mathematics*, **10**(23), 4545.

ZUO Y., MOUZOUN H., DOUDANE S. G., GERDES K. & SAGOT B. (2022). Patent classification using extreme multi-label learning : A case study of french patents. In *SIGIR 2022-PatentSemTech workshop-3rd Workshop on Patent Text Mining and Semantic Technologies*.

A Important Features from Sparse Classifiers

We utilized a Logistic Regression model with TF-IDF sparse features, in which each category corresponds to a classifier. In our data preprocessing step, we first removed stop words and then lemmatized each remaining word. We set the dimensionality of the features to 10,000 and excluded words that occurred in more than 90% of the documents. The most important features are represented by the coefficients in the z-equation of the logistic regression, denoted by w_1 to w_n in the following equation :

$$y = \frac{1}{1 + e^{-z}}$$
$$z = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n$$

To select the most important features for each classifier, we took the top five features with the highest coefficient values. We then identified the 100 words that appeared most frequently across all classifiers and added them to our final encoder. These words will be used to improve the model's accuracy in classification tasks.

title+abstract	**claims**	**description**
composition	outil	moteur
moteur	pourcent	véhicule
outil	composition	outil
composé	moteur	eau
machine	véhicule	composé
roue	formule	polymère
eau	machine	roue
signal	atome	composition
formule	dispositif	fibre
polymère	signal	signal
véhicule	roue	combustion
combustion	groupe	machine
fabrication	couche	gaz
produit	acide	acide
gaz	polymère	air
air	gaz	électrique
commande	air	atome
fibre	eau	mesure
fluide	combustion	fluide
mesure	fibre	piston

TABLE 9 – Examples of important features of logistic regression models trained on different patent parts

B Visualization of IPC distributions

Given that different countries have varying priorities in protecting different technical fields, it is important to consider the distance between training and testing data when performing patent classification. Simply adding more data for training without considering the distribution of the data across countries may lead to suboptimal classification results. To demonstrate this phenomenon, we visualize the distribution of labels at the IPC subclass level in Figure B.

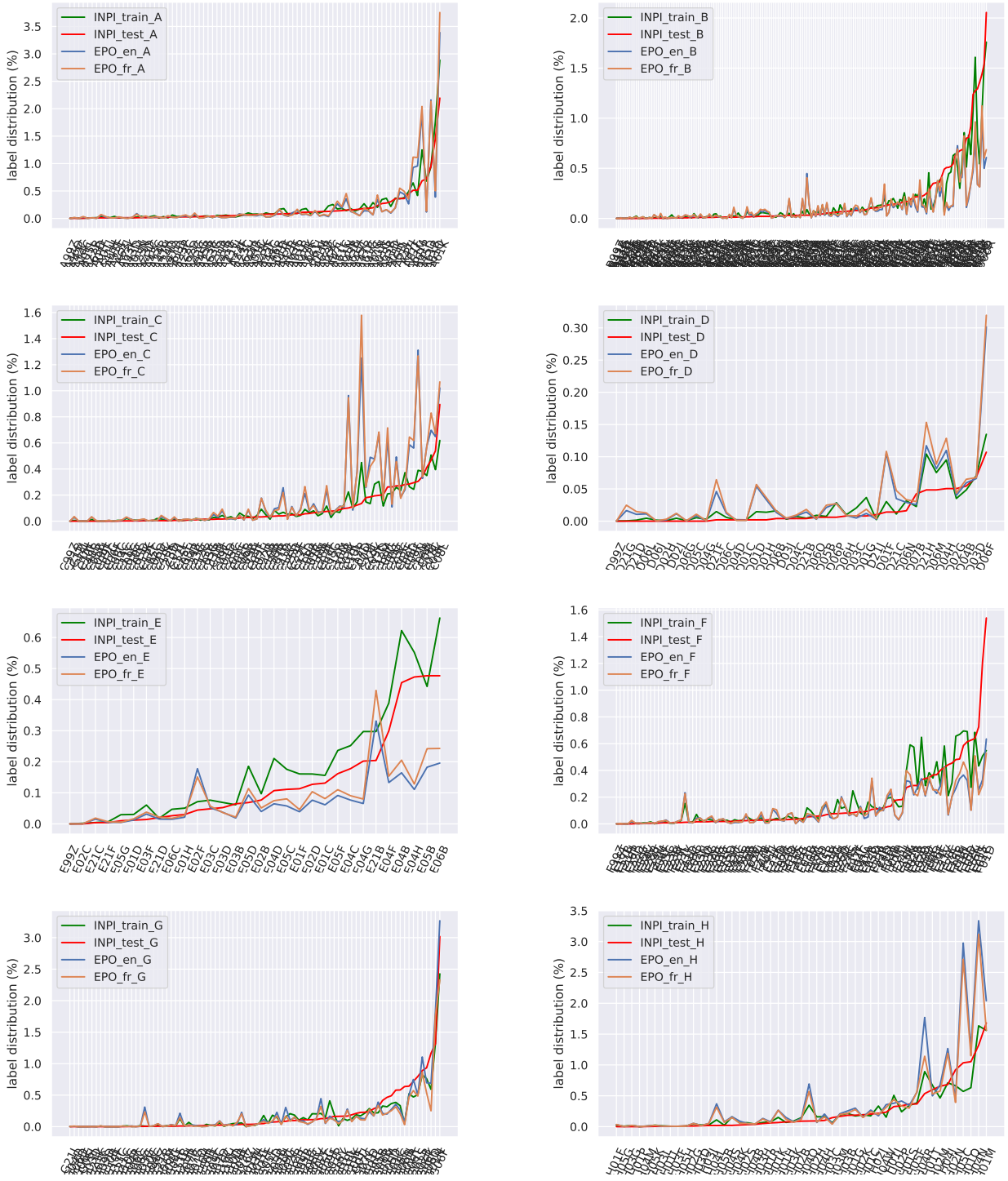


FIGURE 3 – Distribution of labels per IPC section.

C Machine Translation en-fr

In order to improve the classification of French patents, we used a T5 (Raffel *et al.*, 2020) model that was trained on aligned English-French data. Specifically, we leveraged the first release of the EuroPat dataset (Heafield *et al.*, 2022), restricting our analysis to patents that were published after 2010. This allowed us to compile a corpus of 2 million sentence pairs for training our model.

We drew inspiration from (Jehl & Riezler, 2018) to investigate the effectiveness of incorporating special tokens to introducing information as patent sections (<A>, <B>,. . .,<H>) or text types (<title>, <abstract>, <claims>, <description>). Our study compares various methods to determine the optimal approach.

For each approach, we fine-tuned the `t5-base` with 220 million parameters for a single epoch, using a batch size of 16 and a learning rate of 1e-4. The maximum input and output lengths were set to 256. This configuration was selected to optimize the balance between training time and model performance. To evaluate the performance of our model, we relied on the widely used machine translation metric, BLEU score.

	BLEU
Original Text	79.15
IPC1	79.10
Text Type	78.85
IPC1, Text Type	**79.19**

TABLE 10 – Performances of translation models.

We can see that the translation model achieves the best performance when it differentiates between text types and text domains. Therefore, in our main experimental results, we demonstrate the use of the model with special tokens of IPC1s and text types during training for translation and data augmentation.