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 18e Conférence en Recherche d'Information et Applications (CORIA)

Haïfa Zargayouna (Éd.)

Paris, France, 5 au 9 juin 2023

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

Avec le soutien de

Préface

Après Rennes en 2018, l'ARIA (Association francophone de Recherche d'Information et Applications) et l'ATALA (Association pour le Traitement Automatique des Langues) ont décidé de se retrouver à nouveau pour organiser conjointement leur principale conférence.
Au cours de cette manifestation, se tiendront les conférences CORIA (COnférence en Recherche d'Information et Applications) – pour sa dix-huitième édition et TALN (conférence sur le Traitement Automatique des Langues Naturelles) – pour sa trentième édition. À ces deux conférences s'ajoutent les journées jeunes chercheurs, à savoir, les seizième Rencontres Jeunes Chercheurs en RI (RJCRI) et les vingt-cinquième Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL). Ces conférences sont les points de rassemblement des communautés francophones respectivement en recherche d'information et en traitement automatique des langues.

Cette année, le comité de programme de CORIA a choisi 18 papiers pour une présentation orale (10 papiers longs, 2 papiers courts et 6 papiers déjà publiés dans des conférences ou revues internationales de renom). Les papiers couvrent des thèmes de recherche d'information assez divers avec une forte coloration autour des architectures neuronales et des modèles de langue.

Pour favoriser les interactions entre les deux communautés, nous avons intégré dans l'organisation des présentations quatre sessions communes à la recherche d'information et au traitement automatique des langues. Pour compléter le contenu scientifique des conférences, le programme de la manifestation a été enrichi par la venue de trois conférenciers invités dont deux de la communauté de la Recherche d'Information :

— Jaap Kamps est enseignant chercheur à l'Université d'Amsterdam, ses recherches autour de la Recherche d'Information se font à l'Institut pour la logique, le langage et le calcul (Institute for Logic, Language, and Computation).
— Jian-Yun Nie est professeur à Université de Montréal et titulaire d'une chaire de recherche du Canada en traitement de langue naturelle. Il travaille dans le laboratoire RALI (Recherche Appliquée en Linguistique Informatique). Ses recherches couvrent divers sujets en recherche d'information (RI) et traitement de langue naturelle.

Enfin, je tiens à remercier l'ensemble des membres du comité d'orientation et le bureau de l'ARIA, les membres du comité de programme pour l'évaluation des soumissions, les organisateurs Franciliens pour leur grande disponibilité, ainsi que toutes les personnes qui se sont investies pour la réussite de cette manifestation. Je remercie, également, les auteurs grâce à qui CORIA continue d'exister.

Je vous souhaite à toutes et à tous de belles journées scientifiques.

Haïfa Zargayouna – Présidente du Comité de Programme CORIA 2023

Comités

Comité de programme

Présidente

Haïfa ZARGAYOUNA

Membres

— Ismail BADACHE
— Patrice BELLOT
— Catherine BERRUT
— Romaric BESANCON
— Robert BOSSY
— Mohand BOUGHANEM
— Davide BUSCALDI
— Sylvie CALABRETTO
— Max CHEVALIER
— Jean-Pierre CHEVALLET
— Adrian CHIFU
— Vincent CLAVEAU
— Antoine DOUCET
— Sebastien FOURNIER
— Eric GAUSSIER
— Mathias GÉRY
— Lorraine GOEURIOT
— Gilles HUBERT
— Jose MORENO
— Véronique MORICEAU
— Philippe MULHEM
— Diana NURBAKOVA
— Jian-Yun NIE
— Karen PINEL-SAUVAGNAT
— Benjamin PIWOWARSKI
— Mathieu ROCHE
— Eric SAN JUAN
— Jacques SAVOY
— Christophe SERVAN
— Laure SOULIER
— Lynda TAMINE
— Nicolas TURENNE
— Haifa ZARGAYOUNA

Table des matières

I Articles longs 1

Impact de l'apprentissage multi-labels actif appliqué aux transformers 2
Maxime Arens, Charles Teissèdre, Lucile Callebert, Jose G Moreno, Mohand Boughanem

Quelles évolutions sur cette loi ? Entre abstraction et hallucination dans le domaine du résumé de textes juridiques 18
Nihed Bendahman, Karen Pinel-Sauvagnat, Gilles Hubert, Mokhtar Boumedyen Billami

Augmentation de jeux de données RI pour la recherche conversationnelle à initiative mixte 37
Pierre Erbacher, Philippe Preux, Jian-Yun Nie, Laure Soulier

Apprentissage de sous-espaces de préfixes 59
Louis Falissard, Vincent Guigue, Laure Soulier

Recherche cross-modale pour répondre à des questions visuelles 74
Paul Lerner, Ferret Olivier, Camille Guinaudeau

Adaptation de domaine pour la recherche dense par annotation automatique 93
Minghan Li, Eric Gaussier

Extraction d'entités nommées à partir de descriptions d'espèces 111
Maya Sahraoui, Vincent Guigue, Régine Vignes-Lebbe, Marc Pignal

Le théâtre français du XVIIe siècle : une expérience en catégorisation de textes 127
Jacques Savoy

Enrichissement des modèles de langue pré-entraînés par la distillation mutuelle des connaissances 139
Raphaël Sourty, Jose G Moreno, François-Paul Servant, Lynda Tamine

Constitution de sous-fils de conversations d'emails 157
Lionel Tadonfouet Tadjou, Eric De La Clergerie, Fabrice Bourge, Tiphaine Marie

II Articles courts 172

Intégration du raisonnement numérique dans les modèles de langue : État de l'art et direction de recherche 173
Sarah Abchiche, Lynda Said Lhadj, Vincent Guigue, Laure Soulier

Reconnaissance d'Entités Nommées fondée sur des Modèles de Langue Enrichis avec des Définitions des Types d'Entités 185
Jesús Lovón Melgarejo, Jose Moreno, Romaric Besançon, Olivier Ferret, Lynda Tamine

III Articles déjà publiés 195

Entity Enhanced Attention Graph-Based Passages Retrieval 196
Lucas Albarede, Lorraine Goeuriot, Philippe Mulhem, Claude Le Pape-Gardeux, Sylvain Marie, Trinidad

Chardin-Segui

Highlighting exact matching via marking strategies for ad hoc document ranking with pretrained contextualized language models **201**
Lila Boualili, Jose Moreno, Mohand Boughanem

Vers l'évaluation continue des systèmes de recherche d'information. **202**
Petra Galuscakova, Romain Deveaud, Gabriela Gonzalez-Saez, Philippe Mulhem, Lorraine Goeuriot, Florina Piroi, Martin Popel

CoSPLADE : Adaptation d'un Modèle Neuronal Basé sur des Représentations Parcimonieuses pour la Recherche d'Information Conversationnelle **207**
Nam Le Hai, Thomas Gerald, Thibault Formal, Jian-Yun Nie, Benjamin Piwowarksi, Laure Soulier

The Power of Selecting Key Blocks with Local Pre-ranking for Long Document Information Retrieval **213**
Minghan Li, Diana Nicoleta Popa, Johan Chagnon, Yagmur Gizem Cinar, Eric Gaussier

iQPP : Une Référence pour la Prédiction de Performances des Requêtes d'Images **214**
Eduard Poesina, Radu Tudor Ionescu, Josiane Mothe

IV Démonstration **221**

XPMIR : Une bibliothèque modulaire pour l'apprentissage d'ordonnancement et les expériences de RI neuronale **222**
Yuxuan Zong, Benjamin Piwowarski

Première partie

Articles longs

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

Impact de l'apprentissage multi-labels actif appliqué aux transformers

Maxime Arens[1, 2] Charles Teissèdre[2] Lucile Callebert [2] Jose G. Moreno[1]
Mohand Boughanem[1]
(1) Université de Toulouse, IRIT UMR 5505 CNRS, 31400 Toulouse, France
(2) Synapse Développement, 7 Boulevard de la Gare, 31500 Toulouse, France
maxime.arens@irit.fr

RÉSUMÉ

L'Apprentissage Actif (AA) est largement utilisé en apprentissage automatique afin de réduire l'effort d'annotation. Bien que la plupart des travaux d'AA soient antérieurs aux *transformers*, le succès récent de ces architectures a conduit la communauté à revisiter l'AA dans le contexte des modèles de langues pré-entraînés. De plus, le mécanisme de *fine-tuning*, où seules quelques données annotées sont utilisées pour entraîner le modèle sur une nouvelle tâche, est parfaitement en accord avec l'objectif de l'AA. Nous proposons d'étudier l'impact de l'AA dans le contexte des *transformers* pour la tâche de classification multi-labels. Or la plupart des stratégies AA, lorsqu'elles sont appliquées à ces modèles, conduisent à des temps de calcul excessifs, ce qui empêche leurs utilisations au cours d'une interaction homme-machine en temps réel. Afin de pallier ce problème, nous utilisons des stratégies d'AA basées sur l'incertitude. L'article compare six stratégies d'AA basées sur l'incertitude dans le contexte des *transformers* et montre que si deux stratégies améliorent invariablement les performances, les autres ne surpassent pas l'échantillonnage aléatoire. L'étude montre également que les stratégies performantes ont tendance à sélectionner des ensembles d'instances plus diversifiées pour l'annotation.

ABSTRACT

Impact of multi-label active learning applied to transformers

Active Learning (AL) is widely used in machine learning to reduce the human annotation effort. While most AL predates transformers, the recent success of such architectures has led the research community to revisit AL in the context of transformers. Moreover the fine-tuning mechanism, where only few annotated data are used to train the transformer on a downstream task, is fully aligned with the goal of AL. We propose to study the impact of AL in the context of transformers for the multi-label classification task. However, most AL strategies, when applied to transformers, lead to excessive computation times, which prevents their use in real time human-machine interaction. To cope with these efficiency issues, we adopt uncertainty-based AL strategies. The paper compares six uncertainty-based AL strategies on transformers and shows that while two strategies consistently improve performances, the others do not outperform random sampling. The study also shows that these well-performing strategies tend to select batches of more diverse instances for annotation.

MOTS-CLÉS : Apprentissage Actif, classification multi-labels, transformers.

KEYWORDS: Active Learning, multi-label classification, transformers..

1 Introduction

L'acquisition de données annotées est un point central en apprentissage automatique et profond (Fredriksson *et al.*, 2020). En effet, la performance des modèles entraînés est souvent relative à la quantité et la qualité des données annotées à disposition. Le processus d'annotation, particulièrement dans les domaines techniques, est une étape onéreuse qui requiert la participation d'humains voire d'experts (Wu *et al.*, 2021). C'est d'autant plus vrai dans la classification multi-labels (Zhang, 2022) où chaque instance peut avoir plusieurs labels, l'annotation devient alors un processus fastidieux pour l'annotateur, surtout quand l'espace des labels est grand. Le principal objectif de l'Apprentissage Actif (AA) est de réduire le coût d'annotation de données (Wang *et al.*, 2021) en choisissant et en limitant les données à annoter. En effet, au lieu d'annoter de façon aléatoire les données, les stratégies d'AA permettent de sélectionner en priorité les meilleurs ensembles de données à annoter dans le but de maximiser le gain d'information du modèle lors de sa prochaine étape d'entraînement. Les données ainsi sélectionnées sont annotées par un oracle (un humain). Ces deux étapes sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint. Plus la stratégie d'AA est efficace, plus l'interaction avec l'oracle est valorisée.

Bien que les récents développements de nouvelles architectures d'apprentissage profond (Vaswani *et al.*, 2017) aient conduit à quelques travaux étudiant l'utilisation de l'AA sur des *transformers* (Ein-Dor *et al.*, 2020; Lu & MacNamee, 2020; Schröder *et al.*, 2022), la plupart des stratégies d'AA ont été et sont conçues pour des architectures classiques d'apprentissage automatique telles que les machines à vecteurs de support (SVM) ou des réseaux de neurones à convolution (CNN) (Kumar & Gupta, 2020; Schröder & Niekler, 2020; Reyes *et al.*, 2018; Nakano *et al.*, 2020; Gui *et al.*, 2021; Chen *et al.*, 2022). Cela est principalement dû au fait que l'utilisation de l'AA avec des *transformers* entraîne une augmentation du temps de calcul de chaque étape d'entraînement, ce qui rend l'utilisation de ces stratégies peu viables (Schröder *et al.*, 2022). En effet, les stratégies d'AA doivent être efficientes en matière de temps de calculs afin de permettre une interaction humain-machine (Wang *et al.*, 2021). Appliquer l'AA à des *transformers* permet notamment de réduire le coût d'annotation durant le *fine-tuning* (spécialisation d'un modèle sur une nouvelle tâche) en sélectionnant le meilleur ensemble de données à annoter (Ein-Dor *et al.*, 2020; Lu & MacNamee, 2020).

L'objectif de cet article est d'étudier l'impact des stratégies d'AA sur les *transformers*. Nous nous focalisons sur les stratégies basées sur l'incertitude qui ont prouvé leur efficacité sur des architectures de modèles antérieures. Dans le but de généraliser nos résultats, notre étude est réalisée sur quatre jeux de données (Demszky *et al.*, 2020; Chalkidis *et al.*, 2022; Lippi *et al.*, 2018; Chalkidis *et al.*, 2021), utilise une métrique commune dans la classification multi-labels (Tsoumakas *et al.*, 2010) et explore deux modèles : distilBERT et distilRoBERTa (Devlin *et al.*, 2019; Sanh *et al.*, 2019; Liu *et al.*, 2019). La contribution de ce papier est triple :

1. Nous implémentons six stratégies d'AA basées sur l'incertitude sur deux *transformers*.
2. Nos résultats indiquent que *Confidence Minimum No weighting (CMN)* et *Max Margin Uncertainty (MMU)* sont les meilleures stratégies parmi les six étudiées, améliorant les performances des deux modèles sur l'ensemble des jeux de données.
3. Nos résultats montrent que les stratégies performantes ont tendance à sélectionner des ensembles de données plus diverses que les stratégies en sous-performance.

2 Méthodologie

2.1 Contexte

L'enjeu pour entraîner un modèle avec de l'AA consiste en l'élaboration d'une stratégie pour sélectionner une instance plutôt qu'une autre (Settles, 2009). Une fois la stratégie appliquée et l'instance sélectionnée, la plupart des travaux sur l'AA dans la classification multi-labels font ensuite une requête à un oracle afin d'obtenir tous les labels associés à cette instance (Reyes *et al.*, 2018). De récentes études sur l'application de stratégies d'AA à des *transformers*, sur la tâche de classification binaire (Ein-Dor *et al.*, 2020; Lu & MacNamee, 2020) ont montré que l'AA réduisait les biais lors des premières étapes de l'entraînement. Nuançant ce constat, d'autres résultats préliminaires (D'Arcy & Downey, 2022) suggèrent que les stratégies prédatant les *transformers* s'appliquent difficilement à eux, ajoutant de l'instabilité à l'entraînement.

Les stratégies d'AA basées sur les ensembles (Krogh & Vedelsby, 1994; Shi *et al.*, 2011) et les gradients (Ein-Dor *et al.*, 2020; Lu & MacNamee, 2020) s'adaptent mal au grand nombre de paramètres des *transformers* (Schröder *et al.*, 2022). Comme suggéré par (Lu & MacNamee, 2020), pour l'AA dans le contexte des transformers, nous nous concentrons sur l'étude des stratégies basées sur l'incertitude (Lewis & Gale, 1994; Cohn *et al.*, 1996; Li *et al.*, 2004; Esuli & Sebastiani, 2009; Li & Guo, 2013; Reyes *et al.*, 2018).

L'étude la plus complète sur l'utilisation des stratégies d'AA basées sur l'incertitude dans le contexte des *transformers* (Schröder *et al.*, 2022) montre que les stratégies efficaces sur les architectures antérieures (SVM ou CNN) ne sont pas toujours intéressantes pour les *transformers*. Cette étude ne porte pas sur la tâche de classification multi-labels bien qu'elle soit une tâche où l'apport de l'AA est capital (Liu *et al.*, 2021). Comme montré dans (Wertz *et al.*, 2022), certaines stratégies d'AA multi-labels (Reyes *et al.*, 2018; Gissin & Shalev-Shwartz, 2019; Yuan *et al.*, 2020), dans le contexte des *transformers*, ne semblent pas apporter d'améliorations et performent même moins bien qu'un échantillonage aléatoire des données d'entraînement. A notre connaissance, il n'y a pas encore d'explications pour ces résultats et notre article tente de combler ce manquement autour de six stratégies différentes basées sur l'incertitude.

2.2 Stratégies d'Apprentissage Actif multi-labels

La tâche de classification multi-labels consiste à assigner les labels appropriés à des instances textuelles. Contrairement à la classification multi-classes, plusieurs labels peuvent être associés à une même instance. Pour chacune de nos expériences, notre espace de label est prédéfini et n'évolue pas au fur et à mesure. L'objectif de l'AA est de sélectionner les meilleures instances possibles à annoter pour l'entraînement. Cette sélection peut se faire selon différentes *stratégies*. Nos stratégies sont basées sur l'estimation de l'*incertitude* du modèle sur chaque instance, c'est-à-dire la confiance du modèle dans la prédiction des labels associés à cette instance. Ces stratégies reposent sur l'hypothèse qu'en s'entraînant sur des exemples difficiles (où le modèle hésite), le modèle va gagner en performance.

À chaque itération d'entraînement, les stratégies sélectionnent les instances à annoter parmi toutes les données non-annotées du jeu de données (Lewis & Gale, 1994). Pour chaque instance non-annotée, nous calculons un score qui indique l'incertitude du modèle sur ses prédictions associées. Nous appliquons six stratégies d'AA multi-labels aux *transformers* :

Soient nos instances textuelles $x_1, ..., x_n$ et notre espace des labels $l = l^1, ..., l^q$. Pour une instance donnée x_i nous représentons sa distribution probabiliste de labels par $y_i = [y_i^1, ..., y_i^q], y_i^j \in [0, 1]$ où plus y_i^j est proche de 1, plus le modèle est confiant que x_i est labelisé comme j et où plus y_i^j est proche de 0, plus le modèle est confiant que x_i n'est pas labelisé comme j.

Comme suggéré dans (Schröder *et al.*, 2022), nous concentrons notre étude sur des stratégies d'AA basées sur l'incertitude et plus précisement sur six stratégies d'AA multi-labels :

Max Loss (ML) sélectionne les instances pour lesquelles la fonction de perte est la plus élevée (Li *et al.*, 2004) :

$$\underset{x_i}{\operatorname{argmax}} \Big[\sum_{j=1}^{q} \max\{1 - m_j * f_j(x_i), 0\} \Big] \tag{1}$$

où $m_j = 1$ si $j = u$, $m_j = -1$ sinon, u correspondant au label l^u associé avec la plus grande probabilité à une instance donnée et où $f_j(x_i)$ est défini par :

$$f_j(x_i) = 2 * y_i^j - 1 \tag{2}$$

Mean Max Loss (MML) sélectionne les instances pour lesquelles la fonction de perte moyenne est la plus élevée (Li *et al.*, 2004) :

$$\underset{x_i}{\operatorname{argmax}} \frac{1}{q} \Big[\sum_{k=1}^{q} \sum_{j=1}^{q} max\{1 - o_{kj} * f_j(x_i), 0\} \Big] \tag{3}$$

où $o_{kj} = 1$ si $j = k$, $o_{kj} = -1$ sinon et $f_j(x_i)$ est défini dans (2).

Minimum Confidence No weighting (CMN) sélectionne les instances pour lesquelles la confiance du modèle est la plus basse (Esuli & Sebastiani, 2009) :

$$\underset{x_i}{\operatorname{argmin}} \Big(\min_{j=1}^{q} f_j(x_i) \Big) \tag{4}$$

avec $f_j(x_i)$ défini dans (2).

Max Margin Uncertainty sampling (MMU) sélectionne les instances qui maximisent la marge de séparation entre les groupes prédits de labels positifs et négatifs (Li & Guo, 2013) :

$$\underset{x_i}{\operatorname{argmax}} \frac{1}{\min pos(x_i) - \max neg(x_i)} \tag{5}$$

où $pos(x_i) = [pos_1(x_i), ..., pos_q(x_i)]$ et $neg(x_i) = [neg_1(x_i), ..., neg_q(x_i)]$, avec :

$$pos_j(x_i) = \begin{cases} f_j(x_i) & \text{si } f_j(x_i) > 0 \\ +\infty & \text{sinon} \end{cases} \quad \text{et} \tag{6}$$

$$neg_j(x_i) = \begin{cases} f_j(x_i) & \text{si } f_j(x_i) < 0 \\ -\infty & \text{sinon} \end{cases} \tag{7}$$

avec $f_j(x_i)$ défini dans (2).

Label Cardinality Inconsistency (LCI) sélectionne les instances qui maximisent la distance entre le nombre de labels positifs prédits et la cardinalité des labels du jeu de données annotées (Li & Guo, 2013) :

$$\underset{x_i}{\text{argmax}} \sqrt{\left(\left(\sum_{j=1}^{q} y_i^j\right) - L\right)^2} \tag{8}$$

avec L le nombre moyen de labels (cardinalité) sur les instances déjà annotées.

Category Vector Inconsistency and Ranking of Scores (CVIRS) sélectionne les intances suivant deux mesures. La première est basée sur une agrégation de rang des marges de différence des prédictions du classifieur. La seconde est basée sur l'incohérence des ensembles de labels prédits par rapport à l'espace des labels de l'ensemble des instances annotées. Cette stratégie, plus complexe que les autres, est détaillée dans (Reyes *et al.*, 2018).

Ces stratégies sont dites *myopic*, dans le sens où elles évaluent l'incertitude instance par instance. Bien qu'il existe des stratégies prenant en compte la composition des lots (Gui *et al.*, 2021), les travaux appliquant des stratégies d'AA aux *transformers* utilisent principalement des stratégies *myopic*. Comme dans (Reyes *et al.*, 2018), nous adaptons ces stratégies à l'apprentissage par lots de données simplement : au lieu de sélectionner uniquement l'instance pour laquelle le modèle est le plus incertain, nous sélectionnons les instances les plus incertaines pour remplir notre lot d'entraînement.

3 Expérimentation et résultats

Dans nos expérimentations, l'AA multi-labels suit le processus suivant : tout d'abord, nos modèles sont initialisés en les entraînant avec 25 instances sélectionnées aléatoirement. Ensuite, pour chaque stratégie, nous effectuons 50 itérations d'AA où 25 instances sont sélectionnées à chaque itération pour être ensuite annotées par un oracle, ce qui permet de collecter un total de 1250 instances annotées par stratégie. Après chaque itération, nous entraînons à nouveau le modèle avec le nouveau lot d'instances annotées.

Nous comparons les six stratégies d'AA multi-labels (Li *et al.*, 2004; Esuli & Sebastiani, 2009; Li & Guo, 2013; Reyes *et al.*, 2018) sur deux *transformers*, distilBERT et distilRoBERTa (Devlin *et al.*, 2019; Sanh *et al.*, 2019; Liu *et al.*, 2019). Nos expériences sont menées sur quatre jeux de données multi-labels et reportons nos résultats via une métrique largement utilisée afin de généraliser nos résultats. Ces expérimentations ont été réalisées grâce à la librairie *small-text* [1] (Schröder *et al.*, 2023). Les résultats sont une moyenne calculée sur cinq exécutions de chaque expérience, l'écart-type est indiqué sur les graphes.

1. https://small-text.readthedocs.io/en/latest/

TABLEAU 1 – Caractéristiques des jeux de données

Nom	Labels	Entraînement	Test	Cardinalité	Densité
Jigsaw_toxic	6	159,571	63,978	0.222	0.037
Go_emotions	27	43,410	5,427	0.848	0.031
EUR_Lex	100	55,000	5,000	4.526	0.036
UNFAIR-ToS	8	5,532	1,607	0.124	0.016

3.1 Jeux de données

Pour les modèles antérieurs aux *transformers*, l'ensemble de jeux de données référence dans la classification multi-labels était Mulan (Tsoumakas *et al.*, 2011). Les stratégies d'AA que nous étudions ont souvent été évaluées sur cet ensemble. Cependant, comme les instances sont données sous forme de caractéristiques numériques et que le texte original brut n'est pas fourni, ce jeu de données n'est pas approprié pour l'apprentissage des *transformers*.

Le Tableau 1 montre les caractéristiques des quatre jeux de données utilisés. La cardinalité est le nombre moyen de labels par instance. La densité est la cardinalité divisée par le nombre total d'instances.

Nous avons choisi ces jeux de données afin de réaliser nos expériences sur des textes présentant une variation du niveau de langue employé (de l'insulte au texte légal, en passant par du commentaire de réseau social) ainsi qu'une variation du nombre de labels associés à chaque instance (de 6 à 100).

Jigsaw toxic comment classification (Jigsaw_Toxic) est le jeu de données associé à une compétition Kaggle [2] dont le but était de détecter et de classifier six différents types de toxicité que l'on peut trouver en ligne. Les instances sont tirées de commentaires sur des pages wikipédia. Les différents labels, se référant à différents types de toxicité, sont souvent corrélés (par exemple, toutes les instances de "toxicité sévère" sont aussi labellisées "toxicité"). Près de 90% des instances du jeu de données ne présentent aucune forme de toxicité et ne sont donc associées à aucun label.

Go_Emotions est un jeu de données composé de commentaires Reddit [3] labellisés sur 27 catégories d'émotions comme "colère" ou "curieux" (Demszky *et al.*, 2020). Un peu plus de 30% des instances sont "neutres", c'est-à-dire non associées avec un label.

EUR_Lex57K (EUR_Lex) est un jeu de données composé de textes légaux (Chalkidis *et al.*, 2021) issus du site du même nom [4].

UNFAIR - Terms of Services (UNFAIR-ToS) est un jeu de données composé de textes annotés avec huit types de termes contractuels injustes (Lippi *et al.*, 2018), c'est-à-dire, des termes qui violent potentiellement les droits des consommateurs selon la loi de consommation européene.

Dans nos travaux, nous avons utilisé les versions de EUR_Lex et UNFAIR-ToS fournit dans *Legal General Language Understanding Evaluation* (LexGLUE) (Chalkidis *et al.*, 2022).

Dans nos expérimentations, 10% du jeu d'entraînement est utilisé en tant que jeu de validation et les performances reportées sont obtenues à partir du jeu de test.

2. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
3. https://www.reddit.com/
4. https://eur-lex.europa.eu

3.2 Configuration expérimentale

Oracle : La simulation d'un oracle humain, annotant les instances non-annotées sélectionnées par les différentes stratégies, est réalisée par l'utilisation des jeux de données multi-labels annotés. A chaque itération d'entraînement, la stratégie d'AA sélectionne les meilleurs instances à partir du jeu d'entraînement sans avoir accès aux labels correspondants. Ces instances et leurs labels associés composent le prochain lot d'entraînement.

Modèles : Comme dans (Schröder *et al.*, 2022), les deux *transformers* utilisés dans cette études sont basés sur BERT (Devlin *et al.*, 2019) et RoBERTa(Liu *et al.*, 2019). Étant donné que (Tsvigun *et al.*, 2022) montrent que dans les processus d'AA les versions distillées de ces modèles obtiennent des performances similaires à celles obtenues par les modèles originaux, tout en étant moins gourmands en ressources informatiques, nous utilisons également les versions distillées de ces modèles (Sanh *et al.*, 2019). Au-dessus des deux modèles, nous ajoutons une couche de neurones dense ainsi qu'une couche sigmoïde afin de réaliser la classification multi-labels.

Détails d'implémentation : DistilBERT est composé de 6 couches, des unités cachées de taille 768 et de 66G paramètres. DistilRoBERTa est structuré d'une façon comparable, à l'exception de son nombre de paramètres, qui est de 82G. La taille maximum des tokens d'entrées pour les deux modèles est fixée à 128, le nombre d'époques à 15 et la taille des lots d'entraînement à 25. Pour optimiser les paramètres des modèles, nous avons choisi AdamW avec un taux d'apprentissage de 2e-5. Les expériences ont été réalisées sur une Nvidia GTX1080Ti (32 GB). Les mêmes hyper-paramètres ont été utilisés par les deux modèles sur les quatre jeux de données. Nous suivons (Howard & Ruder, 2018) pour notre processus de *fine-tuning*.

Valeurs de références : **Random (RD)** est une référence commune dans l'AA, où les instances à annoter sont échantillonnées de manière aléatoire à partir du jeu de données non-annotées. Afin d'évaluer les performances finales atteintes par les différentes stratégies, nous utilisons comme référence le modèle entraîné sur l'ensemble du jeu de données en supervision totale. Nous nommons cette valeur de référence **Full-Supervision (FULL)**.

Métrique : Pour mesurer la performance, nous utilisons une métrique communément utilisée dans la classification multi-labels (Tsoumakas *et al.*, 2010). Nous utilisons les notations de la section 2.2, en ajoutant : pour un label donné l^j nous notons les vrais positifs (vp^j), les faux positifs (fp^j), les faux négatifs (fn^j) et définissons la F1-mesure comme :

$$F1(vp^j, fp^j, fn^j) = \frac{vp^j}{vp^j + \frac{1}{2}(fp^j + fn^j)} \tag{9}$$

Nous utilisons une F1-mesure avec une micro-moyenne, c'est-à-dire que nous faisons la somme de tous les vrais positifs, faux positifs et faux négatifs pour tous les labels puis nous calculons la F1-mesure (plus la valeur est haute plus la performance est bonne) :

$$M_{iF1} = F_1(\sum_{j=1}^{q} vp^j, \sum_{j=1}^{q} fp^j, \sum_{j=1}^{q} fn^j) \tag{10}$$

3.3 Analyse des données sélectionnées

Pour mieux comprendre comment certains biais de sélection peuvent avoir un impact sur les performances d'une stratégie d'AA, nous avons examiné plusieurs caractéristiques des instances : la taille et la cardinalité, ainsi que certaines caractéristiques des lots : la présence d'aberrations et la similarité des instances. Lorsque l'on adapte des stratégies d'AA *myopic* à de l'apprentissage par lot, il y a un risque d'avoir au sein d'un même lot des instances vecteurs d'informations redondantes (Gui *et al.*, 2021). Cette redondance d'information au sein des lots étant proche conceptuellement de la similarité des instances au sein d'un lot, nous nous concentrons sur cette caractéristique.

Afin de calculer la similarité entre les instances sélectionnées, nous calculons Sim_batch, selon la méthode détaillée dans l'Algorithme 1 : pour commencer nous calculons les plongements lexicaux de chaque instance dans le lot grâce à SentenceTransformer (Reimers & Gurevych, 2019) [5]. Ensuite, nous faisons la moyenne pour chaque lot de la similarité cosinus par paire entre les plongements lexicaux présents dans le lot. Enfin, nous calculons Sim_batch, la moyenne du score de similarité sur tous les lots.

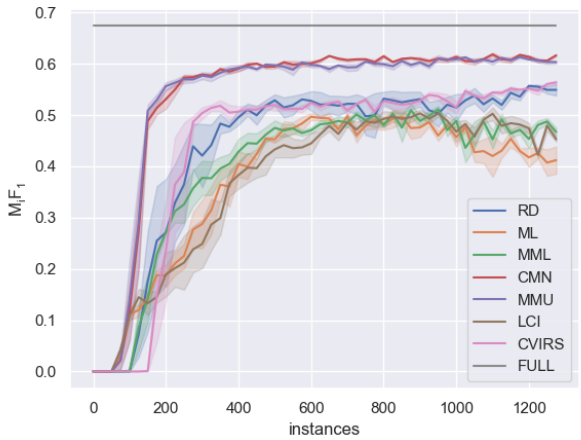
Algorithme 1 Calcul du score Sim_batch

Soit : jeu_annote la liste des lots de données annotées accumulées au fur et à mesure de l'exécution des stratégies d'apprentissage actif
$moyenne\_lot \leftarrow [\,]$
pour chaque lot dans jeu_annote **:**
$pl \leftarrow [\,]$
pour chaque $instance$ dans lot **:**
$pl.ajouter(SentenceTransformer(instance))$
fin pour chaque
$moyenne\_lot.ajouter(\frac{\sum_{x \in paires} similarite\_cosinus(x)}{\frac{|pl|*(|pl|-1)}{2}})$, avec $paires$ la liste des paires uniques d'éléments $\in pl$
fin pour chaque
$Sim\_batch = \frac{\sum_{x \in moyenne\_lot} x}{|moyenne\_lot|}$
retourner Sim_batch

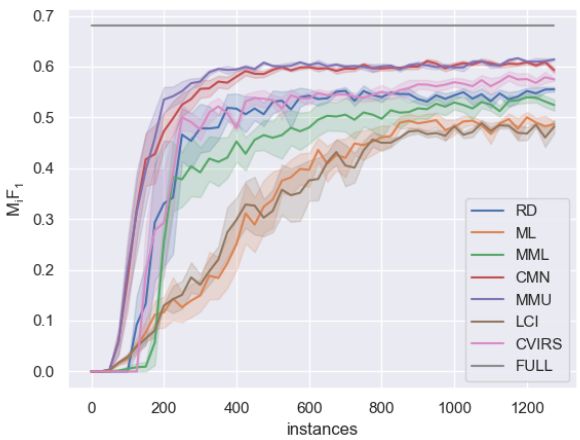
3.4 Résultats

La Figure 1 montre les courbes d'apprentissage en M_{iF1} de nos deux *transformers* selon chaque stratégie d'AA sur les différents jeux de données. Ces résultats indiquent que les stratégies d'AA peuvent améliorer de façon significative les performances atteintes, s'approchant d'une supervision complète à un coût moindre d'annotations. En effet, pour atteindre les performances de l'échantillonnage aléatoire après sélection de 1250 instances, MMU nécessite seulement 175 instances pour distilBERT (225 pour distilRoBERTa) et CMN 175 instances pour distilBERT (275 pour disilRoBERTa). L'écart-type des performances au cours des différentes exécutions est faible pour chaque stratégie d'AA sur les différents jeux de données. En effet, la "graine aléatoire" ne détermine que la composition du lot de

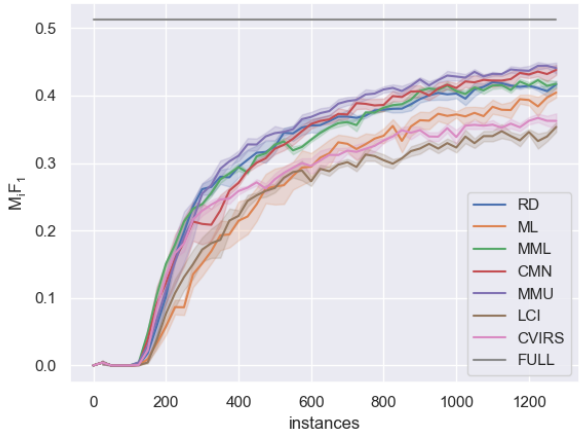
5. nous avons utilisé 'sentence-transformers/nli-distilroberta-base-v2' sur distilRoBERTa et 'sentence-transformers/nli-distilbert-base' sur distilBert



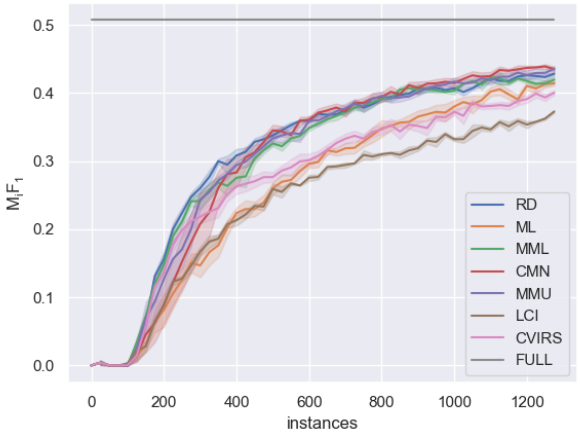
(a) distilBERT sur Jigsaw_toxic



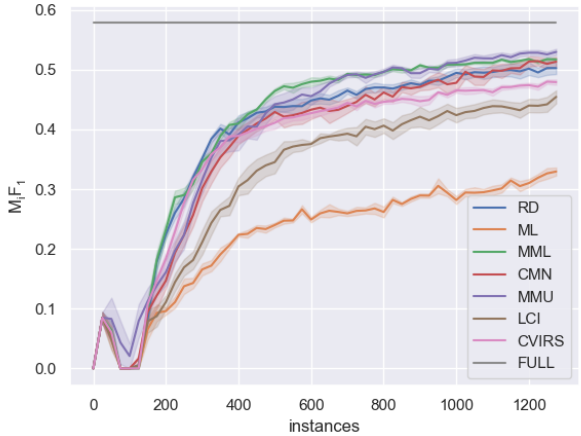
(b) distilRoBERTa sur Jigsaw_toxic



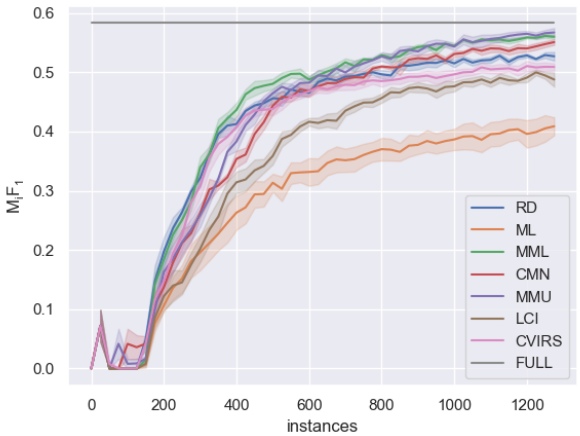
(c) distilBERT sur Go_emotions



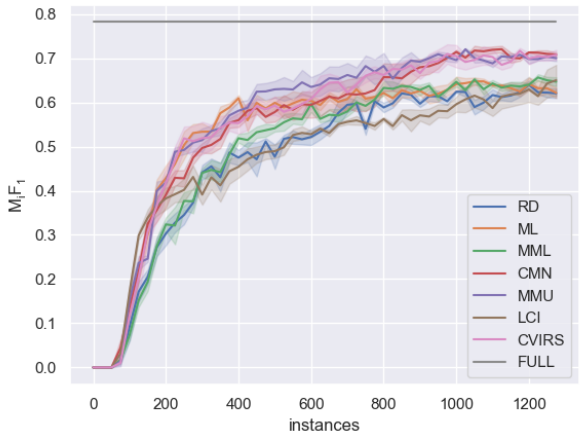
(d) distilRoBERTa sur Go_emotions



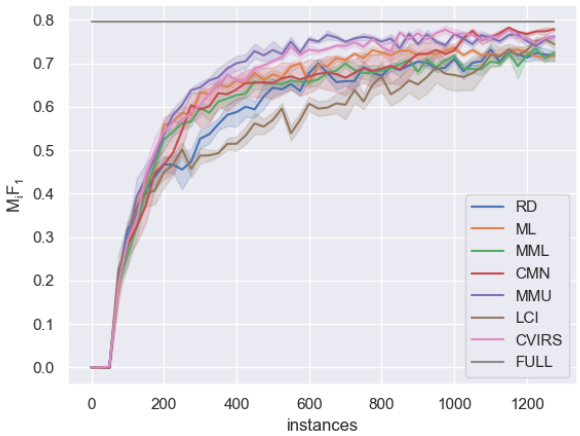
(e) distilBERT sur EUR_Lex



(f) distilRoBERTa sur EUR_Lex



(g) distilBERT sur UNFAIR-ToS



(h) distilRoBERTa sur UNFAIR-ToS

FIGURE 1 – Performances M_{iF1} suivant les différentes stratégies d'AA pour chaque *transformers*

TABLEAU 2 – Pourcentage des jeux de données annotés, associés aux pourcentages de performances atteintes par les stratégies par rapport à une supervision complète.

Jeux de données	Modèles	% jeu de données	% performances M_{iF1}		
			RD	CMN	MMU
Jigsaw	distilBert	0.78	81.45	91.54	89.47
	distilRoBERTa	0.78	81.62	87.21	90.29
goemotions	distilBert	2.88	81.25	85.55	86.13
	distilRoBERTa	2.88	82.84	85.80	85.80
eurlex	distilBert	2.27	86.28	88.00	90.9
	distilRoBERTa	2.27	91.52	95.16	98.1
unfairtos	distilBert	22.60	79.28	90.54	89.51
	distilRoBERTa	22.60	90.19	97.86	95.85

TABLEAU 3 – "M_{iF1}/Sim_batch", obtenu suivant chaque stratégies d'AA. Les résultats en rouge indiquent un score M_{iF1} inférieur à l'échantillonnage aléatoire, les résultats en bleu indiquent un score M_{iF1} supérieur à l'échantillonnage aléatoire.

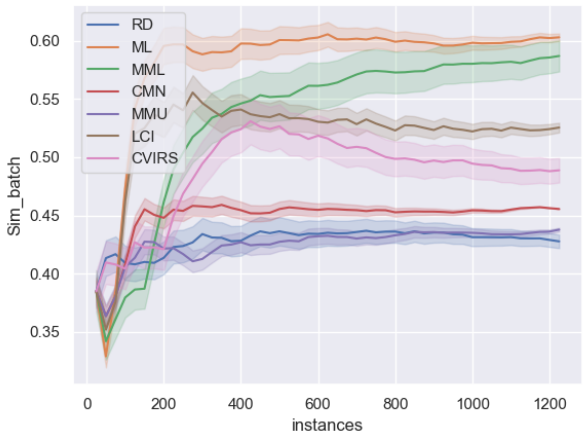
Jeux de données	Modèles	RD	ML	MML	CMN	MMU	LCI	CVIRS	FULL
Jigsaw_toxic	distilBERT	0.549/0.426	0.412/0.602	0.467/0.588	0.617/0.455	0.603/0.439	0.453/0.526	0.564/0.487	0.674/-
	distilRoBERTa	0.555/0.430	0.486/0.590	0.524/0.592	0.593/0.461	0.614/0.448	0.482 /0.578	0.575/0.498	0.680/-
Go_emotions	distilBERT	0.416/0.336	0.404/0.456	0.417/0.341	0.438/0.383	0.441/0.377	0.353/0.474	0.362/0.426	0.512/-
	distilRoBERTa	0.420/0.339	0.414/0.453	0.428/0.341	0.435/0.369	0.435/0.368	0.373/0.463	0.400/0.419	0.507/-
EUR_Lex	distilBERT	0.503/0.741	0.329/0.788	0.517/0.740	0.513/0.736	0.530/0.738	0.454/0.752	0.479/0.758	0.583/-
	distilRoBERTa	0.529/0.736	0.409/0.781	0.560/0.739	0.550/0.746	0.567/0.742	0.488/0.765	0.509/0.756	0.578/-
UNFAIR-ToS	distilBERT	0.620/0.454	0.622/0.490	0.647/0.491	0.708/0.522	0.700/0.504	*0.650*/0.497	0.708/0.523	0.782/-
	distilRoBERTa	0.717/0.455	0.718/0.512	0.724/0.492	0.778/0.553	0.762/0.535	0.744/0.539	0.760/0.511	0.795/-

données d'initialisation dans toutes les stratégies, à l'exception de RD dans laquelle elle joue un rôle plus important.

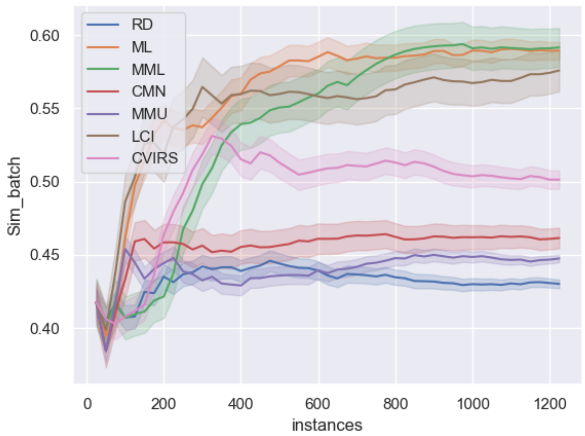
Dans le Tableau 2 nous indiquons le pourcentage du jeu de données que représentent les 1250 instances annotées et le pourcentage des performances de FULL qu'atteignent CMN et MMU, les deux stratégies d'AA les plus performantes, comparativement à l'échantillonnage aléatoire RD. Par exemple pour Jigsaw_Toxic, nous voyons qu'avec moins de 1% du jeu de données (0,78%), nous atteignons plus de 90% des performances de la supervision complète pour CMN (91,54% pour distilBERT et 90,29% pour distilRoBERTa), contre seulement 81% de ces performances avec RD (81,45% pour distilBERT et 81,62% pour distilRoBERTa). De plus, nous voyons sur la Figure 1a que CMN et MMU atteignent ces performances autour de seulement 400 données sélectionnées. Ces résultats mettent en évidence les gains potentiels liés à l'application de stratégies d'AA performantes sur les *transformers*.

Le Tableau 3 compare les performances obtenues après entraînement sur 1250 instances annotées pour chaque combinaison de modèle, jeu de données et stratégies. Afin de mieux illustrer la relation entre les performances du modèle et la similarité des instances sélectionnées, nous avons ajouté le score Sim_batch à côté de la métrique de performance (M_{iF1}/Sim_batch).

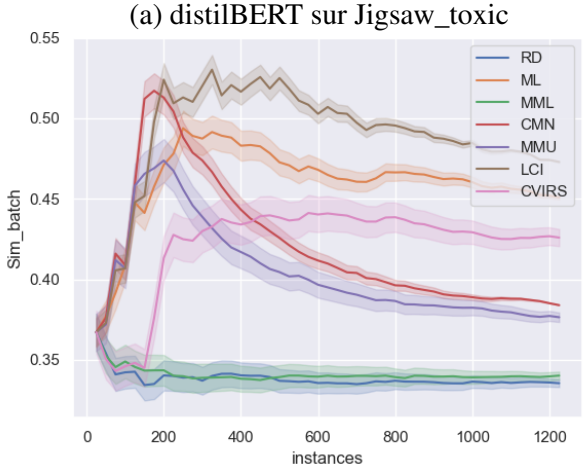
L'un des résultats que nous pouvons tirer du Tableau 3 est qu'à la fin de l'entraînement, la performance (en terme de M_{iF1}) des *transformers* dépend de la stratégie d'AA. Cela suggère que l'ordre des instances sur lequel les *transformers* s'entraînent a bien une importance. Pour les deux *transformers*, CMN and MMU surpassent constamment l'échantillonnage aléatoire et toutes les autres stratégies



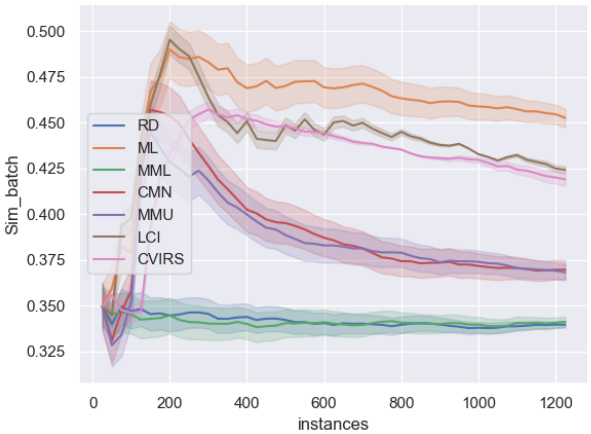
(a) distilBERT sur Jigsaw_toxic



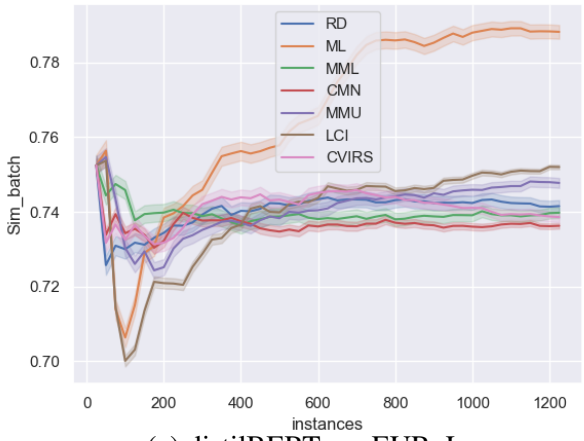
(b) distilRoBERTa sur Jigsaw_toxic



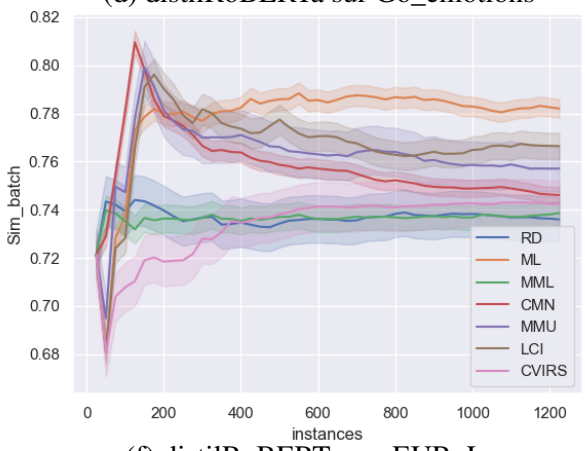
(c) distilBERT sur Go_emotions



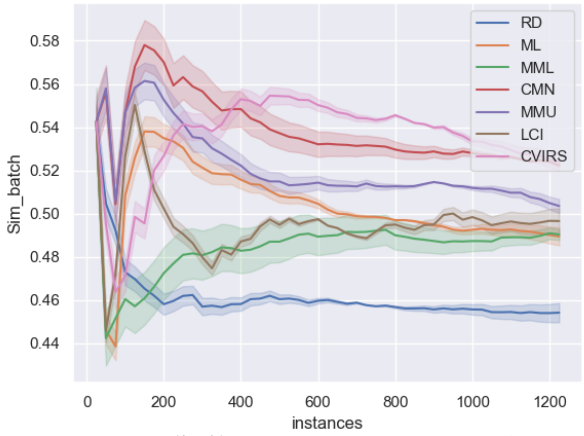
(d) distilRoBERTa sur Go_emotions



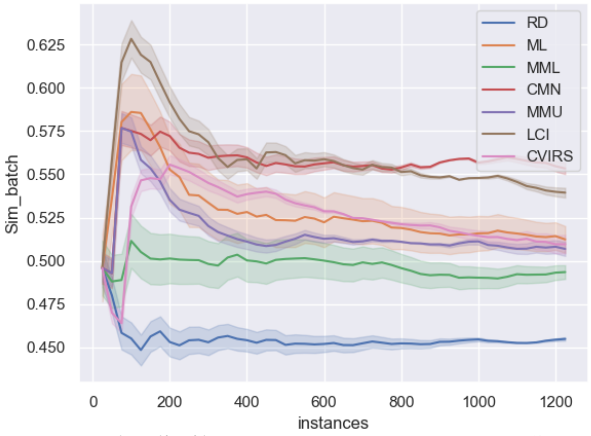
(e) distilBERT sur EUR_Lex



(f) distilRoBERTa sur EUR_Lex



(g) distilBERT sur UNFAIR-ToS



(h) distilRoBERTa sur UNFAIR-ToS

FIGURE 2 – *Sim_batch* suivant les différentes stratégies d'AA pour chaque *transformers*

TABLEAU 4 – Corrélation de Pearson entre M_{iF1} et *Sim_batch*

Jeux de données	Modèles	Pearson	p-value
Jigsaw_toxic	distilBERT	-0.877	0.0096
	distilRoBERTa	-0.843	0.0170
Go_emotions	distilBERT	-0.781	0.0381
	distilRoBERTa	-0.769	0.0431
EUR_Lex	distilBERT	-0.970	0.0003
	distilRoBERTa	-0.930	0.0024
UNFAIR-ToS	distilBERT	0.863	0.0123
	distilRoBERTa	0.776	0.0419

AA, se rapprochant le plus de FULL sur tous les jeux de données.

Une seconde observation que nous pouvons faire, est qu'à chaque fois qu'une stratégie d'AA performe moins bien que l'échantillonnage aléatoire, le score Sim_batch est significativement plus élevé que pour les stratégies performantes ou que l'échantillonnage aléatoire. Dans le Tableau 4, nous remarquons une corrélation linéaire (corrélation de Pearson) entre M_{iF1} et Sim_batch. En effet, il y a une corrélation statistiquement significative (p-value<0.05) entre les performances des stratégies d'AA et le score *Sim_batch*. Plus les instances sélectionnées sont similaires, plus la performance obtenue est basse, à l'exception du jeu de donnée UNFAIR-ToS, pour lequel la corrélation est inversée.

Afin d'expliquer la corrélation obtenue sur les trois premiers jeux de données, en sélectionnant des instances similaires, certaines stratégies entraînent le modèle sur des informations redondantes. La Figure 2, montre l'évolution du score *Sim_batch* au fil de l'apprentissage. Nous remarquons une grande augmentation de score Sim_batch en début d'expérience pour de nombreuses stratégies. Lorsque nous regardons en détail les lots associés à ces fortes augmentations, nous constatons la présence de nombreuses paires de données similaires et même pratiquement identiques. Lorsque ces paires de données très similaires intègrent les lots d'entraînement, cela entraîne un effet domino avec une proportion de paires similaires au sein des lots de plus en plus importante. En poussant plus loin cette analyse, nous avons remarqué ce qui différencie les stratégies performantes des autres : lorsque cet effet domino s'enclenche, CMN et MMU parviennent à en sortir plus rapidement que les autres stratégies. Nous remarquons en effet, que les courbes de Sim_batch pour CMN et MMU ont l'allure d'un pic tandis que les courbes des autres stratégies ont plutôt l'allure d'un pallier.

Pour expliquer la corrélation inversée sur UNFAIR-ToS, un jeu de données composé de phrases annotées de huit différents types de termes contractuels injustes, nous pensons que la similarité des instances joue un rôle clé pour comparer des contrats proches et identifier les parties injustes. Cela peut aussi expliquer pourquoi UNFAIR-ToS est le seul jeu de données pour lequel toutes les stratégies d'AA performent mieux que l'échantillonnage aléatoire.

4 Conclusion

Cet article évalue l'impact sur les *transformers* de six stratégies d'apprentissage actifs basées sur l'incertitude. Nous avons montré que deux de ces stratégies, CMN et MMU, fournissent un gain de performance constant et substantiel par rapport à l'échantillonnage aléatoire, soulignant l'utilité de l'AA appliquée aux *transformers*. Ces deux stratégies seraient notamment de bonnes références pour des prochains travaux sur l'AA multi-labels dans le contexte des *transformers*. Les quatre autres

stratégies fournissent des résultats équivalents ou inférieurs à l'échantillonnage aléatoire. Nous avons investigué les raisons possibles de ces différences de performances. Nous avons d'abord trouvé que CMN et MMU sélectionnent en moyenne des lots d'instances avec une diversité textuelle plus grande que ceux sélectionnés par les stratégies en sous-performance. En poussant notre analyse, nous avons ensuite mis en évidence qu'en début d'expériences, certaines stratégies d'AA sélectionnent des lots de données avec de plus en plus de paires d'instances similaires. CMN et MMU se différenciant par le fait qu'elles arrivent mieux à re-sélectionner des lots avec de moins en moins de paires d'instances similaires dans la suite des expériences. Sur la base de ces résultats, nous nous concentrerons à l'avenir sur des stratégies qui prennent en compte la composition du lot pendant les processus d'AA afin de limiter la présence d'informations redondantes. Par exemple, en maximisant la diversité des instances au sein des lots sélectionnés, en sélectionnant des instances représentatives de l'ensemble du jeu données ou en empêchant la présence de paires d'instances similaires au sein d'un même lot. De plus, nous tenterons de confirmer notre intuition, que les *transformers* sont particulièrement sensibles à la présence d'information redondante dans les lots d'entraînements comparativement à des architectures de modèles antérieurs.

Références

CHALKIDIS I., FERGADIOTIS M. & ANDROUTSOPOULOS I. (2021). MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. *CoRR*, **abs/2109.00904**.

CHALKIDIS I., JANA A., HARTUNG D., BOMMARITO M., ANDROUTSOPOULOS I., KATZ D. M. & ALETRAS N. (2022). LexGLUE : A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dubln, Ireland.

CHEN S., WANG R., LU J. & WANG X. (2022). Stable matching-based two-way selection in multi-label active learning with imbalanced data. *Information Sciences*, **610**, 281–299. DOI : https ://doi.org/10.1016/j.ins.2022.07.182.

COHN D. A., GHAHRAMANI Z. & JORDAN M. I. (1996). Active learning with statistical models. *J. Artif. Int. Res.*, **4**(1), 129–145.

D'ARCY M. & DOWNEY D. (2022). Limitations of active learning with deep transformer language models. URL : https://openreview.net/forum?id=Q8OjAGkxwP5.

DEMSZKY D., MOVSHOVITZ-ATTIAS D., KO J., COWEN A., NEMADE G. & RAVI S. (2020). GoEmotions : A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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.

EIN-DOR L., HALFON A., GERA A., SHNARCH E., DANKIN L., CHOSHEN L., DANILEVSKY M., AHARONOV R., KATZ Y. & SLONIM N. (2020). Active Learning for BERT : An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 7949–7962, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.638.

ESULI A. & SEBASTIANI F. (2009). Active learning strategies for multi-label text classification. In M. BOUGHANEM, C. BERRUT, J. MOTHE & C. SOULE-DUPUY, Éds., *Advances in Information Retrieval*, p. 102–113, Berlin, Heidelberg : Springer Berlin Heidelberg.

FREDRIKSSON T., MATTOS D. I., BOSCH J. & OLSSON H. H. (2020). Data labeling : An empirical investigation into industrial challenges and mitigation strategies. In M. MORISIO, M. TORCHIANO & A. JEDLITSCHKA, Éds., *Product-Focused Software Process Improvement*, p. 202–216, Cham : Springer International Publishing.

GISSIN D. & SHALEV-SHWARTZ S. (2019). Discriminative active learning. *CoRR*, **abs/1907.06347**.

GUI X., LU X. & YU G. (2021). Cost-effective batch-mode multi-label active learning. *Neurocomputing*, **463**, 355–367. DOI : https ://doi.org/10.1016/j.neucom.2021.08.063.

HOWARD J. & RUDER S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 328–339, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1031.

KROGH A. & VEDELSBY J. (1994). Neural network ensembles, cross validation, and active learning. In G. TESAURO, D. TOURETZKY & T. LEEN, Éds., *Advances in Neural Information Processing Systems*, volume 7 : MIT Press.

KUMAR P. & GUPTA A. (2020). Active learning query strategies for classification, regression, and clustering : A survey. *Journal of Computer Science and Technology*, **35**(4), 913–945. DOI : 10.1007/s11390-020-9487-4.

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.

LI X. & GUO Y. (2013). Active learning with multi-label svm classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, p. 1479–1485 : AAAI Press.

LI X., WANG L. & SUNG E. (2004). Multilabel svm active learning for image classification. In *2004 International Conference on Image Processing,2004. ICIP '04.*, volume 4, p. 2207–2210 Vol. 4. DOI : 10.1109/ICIP.2004.1421535.

LIPPI M., PALKA P., CONTISSA G., LAGIOIA F., MICKLITZ H., SARTOR G. & TORRONI P. (2018). CLAUDETTE : an automated detector of potentially unfair clauses in online terms of service. *CoRR*, **abs/1805.01217**.

LIU W., WANG H., SHEN X. & TSANG I. (2021). The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01), 1–1. DOI : 10.1109/TPAMI.2021.3119334.

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. DOI : 10.48550/ARXIV.1907.11692.

LU J. & MACNAMEE B. (2020). Investigating the effectiveness of representations based on pretrained transformer-based language models in active learning for labelling text datasets. *CoRR*, **abs/2004.13138**.

NAKANO F. K., CERRI R. & VENS C. (2020). Active learning for hierarchical multi-label classification. *Data Mining and Knowledge Discovery*, **34**(5), 1496–1530. DOI : 10.1007/s10618-020-00704-w.

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* : Association for Computational Linguistics.

REYES O., MORELL C. & VENTURA S. (2018). Effective active learning strategy for multi-label learning. *Neurocomputing*, **273**, 494–508. DOI : https ://doi.org/10.1016/j.neucom.2017.08.001.

SANH V., DEBUT L., CHAUMOND J. & WOLF T. (2019). DistilBERT, a distilled version of bert : smaller, faster, cheaper and lighter. *ArXiv*, **abs/1910.01108**.

SCHRÖDER C., MÜLLER L., NIEKLER A. & POTTHAST M. (2023). Small-text : Active learning for text classification in python. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics : System Demonstrations*, p. 84–95, Dubrovnik, Croatia : Association for Computational Linguistics.

SCHRÖDER C. & NIEKLER A. (2020). A survey of active learning for text classification using deep neural networks. *CoRR*, **abs/2008.07267**.

SCHRÖDER C., NIEKLER A. & POTTHAST M. (2022). Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics : ACL 2022*, p. 2194–2203, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.findings-acl.172.

SETTLES B. (2009). Active learning literature survey. *Technical Report TR-1648. University of Wisconsin-Madison. Department of Computer Sciences*.

SHI C., KONG X., YU P. S. & WANG B. (2011). Multi-label ensemble learning. In D. GUNOPULOS, T. HOFMANN, D. MALERBA & M. VAZIRGIANNIS, Éds., *Machine Learning and Knowledge Discovery in Databases*, p. 223–239, Berlin, Heidelberg : Springer Berlin Heidelberg.

TSOUMAKAS G., KATAKIS I. & VLAHAVAS I. (2010). *Mining Multi-label Data*, In O. MAIMON & L. ROKACH, Éds., *Data Mining and Knowledge Discovery Handbook*, p. 667–685. Springer US : Boston, MA. DOI : 10.1007/978-0-387-09823-4_34.

TSOUMAKAS G., SPYROMITROS-XIOUFIS E., VILCEK J. & VLAHAVAS I. (2011). Mulan : A java library for multi-label learning. *Journal of Machine Learning Research*, **12**, 2411–2414.

TSVIGUN A., SHELMANOV A., KUZMIN G., SANOCHKIN L., LARIONOV D., GUSEV G., AVETISIAN M. & ZHUKOV L. (2022). Towards computationally feasible deep active learning. In *Findings of the Association for Computational Linguistics : NAACL 2022*, p. 1198–1218, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.findings-naacl.90.

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 Z. J., CHOI D., XU S. & YANG D. (2021). Putting humans in the natural language processing loop : A survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, p. 47–52, Online : Association for Computational Linguistics.

WERTZ L., MIRYLENKA K., KUHN J. & BOGOJESKA J. (2022). Investigating active learning sampling strategies for extreme multi label text classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 4597–4605, Marseille, France : European Language Resources Association.

WU X., XIAO L., SUN Y., ZHANG J., MA T. & HE L. (2021). A survey of human-in-the-loop for machine learning. *ArXiv*, **abs/2108.00941**.

YUAN M., LIN H.-T. & BOYD-GRABER J. (2020). Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 7935–7948, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.637.

ZHANG J. (2022). Knowledge learning with crowdsourcing : A brief review and systematic perspective. *IEEE/CAA Journal of Automatica Sinica*, **9**(5), 749–762. DOI : 10.1109/jas.2022.105434.

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

Quelles évolutions sur cette loi ? Entre abstraction et hallucination dans le domaine du résumé de textes juridiques

Nihed Bendahman[1, 2] Karen Pinel-Sauvagnat[1] Gilles Hubert[1]
Mokhtar Boumedyen Billami[2]
(1) IRIT, 118 Route de Narbonne, 31400 Toulouse, France
(2) Berger-Levrault, 64 Rue Jean Rostand, 31670 Labège, France
{nihed.bendahman, karen.sauvagnat, gilles.hubert}@irit.fr
{nihed.bendahman, mb.billami}@berger-levrault.com

RÉSUMÉ

Résumer automatiquement des textes juridiques permettrait aux chargés de veille d'éviter une surcharge informationnelle et de gagner du temps sur une activité particulièrement chronophage. Dans cet article, nous présentons un corpus de textes juridiques en français associés à des résumés de référence produits par des experts, et cherchons à établir quels modèles génératifs de résumé sont les plus intéressants sur ces documents possédant de fortes spécificités métier. Nous étudions quatre modèles de l'état de l'art, que nous commençons à évaluer avec des métriques traditionnelles. Afin de comprendre en détail la capacité des modèles à transcrire les spécificités métiers, nous effectuons une analyse plus fine sur les entités d'intérêt. Nous évaluons notamment la couverture des résumés en termes d'entités, mais aussi l'apparition d'informations non présentes dans les documents d'origine, dites hallucinations. Les premiers résultats montrent que le contrôle des hallucinations est crucial dans les domaines de spécialité, particulièrement le juridique.

ABSTRACT

What are the evolutions of this law ? Between abstraction and hallucination in the field of legal text summarization

Automatically summarizing legal texts would allow monitoring officers to avoid information overload and to save time on a particularly time-consuming activity. In this paper, we present a corpus of French legal texts associated with reference summaries produced by experts. Using this collection, we aim at identifying which generative summarization models are the most interesting on these documents with important specificities. We study four state-of-the-art models, which we begin to evaluate with traditional metrics. In order to understand in detail the ability of this models to transcribe documents specificities, we perform a more detailed analysis on the entities of interest. In particular, we evaluate the coverage of the produced summaries in terms of entities, but also the appearance of information not present in the original documents, called hallucinations. First results show that hallucination control is crucial in specific domains such as the legal one.

MOTS-CLÉS : Résumés abstractifs, Hallucination, Evaluation, Domaine juridique.

KEYWORDS: Abstractive Summarization, Hallucination, Evaluation, Legal domain.

Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA), pages 18–36.

1 Introduction

La veille juridique est une activité cruciale pour les entreprises afin de rester en phase avec l'actualité juridique. Elle leur permet d'être en permanence au fait des réglementations en cours et d'anticiper leurs évolutions à venir, afin de les appliquer au plus tôt. Cependant, avec l'inflation législative permanente, les chargés de veille éprouvent eux-mêmes une surcharge informationnelle qui complique leurs activités. Il devient très difficile pour eux d'analyser des centaines voire des milliers d'articles par jour. La synthèse de l'information qualifiée de pertinente requiert un effort considérable : identifier dans les amas de textes et de médias les éléments informationnels qui ont réellement de l'importance est particulièrement chronophage. Les chargés de veille travaillent ensuite par extraction et par abstraction de l'information pour construire des résumés dans des newsletters qui soient synthétiques et digestes.

La génération automatique de résumé représente donc une solution intéressante pour aider les chargés de veille dans leurs activités de veille juridique. Les approches de résumé peuvent être « extractives » ou « abstractives ». Les approches extractives, comme (Fabbri *et al.*, 2019; Saini *et al.*, 2019; Zhong *et al.*, 2020), retournent des extraits des textes à résumer, tandis que les approches abstractives, comme (Qi *et al.*, 2020; Zhang *et al.*, 2020a; Dou *et al.*, 2021), peuvent formuler de nouvelles phrases. Les approches abstractives visent donc à produire des résumés analogues à ce que produisent les chargés de veille.

À l'instar d'autres tâches liées au Traitement Automatique des Langues (TAL), les approches récentes de résumé automatique utilisent des modèles de langue neuronaux. Ces approches ont été essentiellement appliquées sur des collections d'actualités (« news ») (Dernoncourt *et al.*, 2018; Ma *et al.*, 2022). À notre connaissance, aucune étude de ce type d'approche n'a été réalisée dans le contexte d'informations juridiques, qui plus est en langue française.

Cet article vise donc à étudier dans quelle mesure les modèles de langue peuvent être appliqués à du résumé abstractif dans le cadre d'informations juridiques. Plus précisément, nous cherchons à répondre aux questions de recherche suivantes :

RQ1. Quelle collection de documents juridiques en français peut être utilisée pour une telle étude ?

RQ2. Quelles sont les performances atteignables par les modèles génératifs de l'état de l'art ? Quels sont les modèles qui fonctionnent le mieux ?

RQ3. Dans quelle mesure les métriques traditionnelles permettent-elles d'interpréter correctement les résultats d'évaluation dans un contexte métier ?

Pour répondre à chacune de ces questions, les contributions de cet article sont :

C1. L'identification d'une collection de test sur le juridique en langue française adaptée à l'évaluation de modèles génératifs de langue pour le résumé abstractif,

C2. La comparaison de modèles de l'état de l'art suivant les familles de métriques traditionnellement reportées (El-Kassas *et al.*, 2021; Ermakova *et al.*, 2019), c'est-à-dire ROUGE et BLEU, complétées par une similarité sémantique,

C3. L'utilisation de métriques basées sur les entités et les mots-clés relatifs à un domaine métier, notamment pour évaluer la couverture des résumés et l'apparition d'informations non présentes dans les documents d'origine, dites « hallucinations » (Akani *et al.*, 2022).

L'article est organisé comme suit. La section 2 présente une synthèse des travaux relatifs à la génération automatique de résumé abstractif et son évaluation. La section 3 présente la collection de documents juridiques en langue française permettant d'évaluer les performances des modèles génératifs de l'état de l'art. Dans la section 4, nous présentons le cadre expérimental puis détaillons et analysons les résultats suivant les métriques ROUGE, BLEU et de similarité sémantique . La section 5 détaille ensuite les résultats des évaluations suivant les entités d'intérêt. Enfin, la section 6 conclut l'article et annonce les pistes de travaux futurs.

2 État de l'art

2.1 Résumés abstractifs

Les approches de génération de résumé abstractif ont toujours été au cœur des recherches en traitement automatique des langues, car elles visent à produire des résumés de texte qui sont plus fluides et plus lisibles que les résumés extractifs. Tandis que les résumés extractifs sélectionnent des phrases ou des passages clés du document source, les résumés abstractifs quant à eux reformulent ce dernier de sorte à exprimer son essence.

Les premières approches neuronales de résumé abstractif sont basées sur des réseaux de neurones récurrents (RNN) (Rumelhart *et al.*, 1986) et leurs variantes LSTM (Hochreiter & Schmidhuber, 1997) et bi-LSTM (Huang *et al.*, 2015). Ces approches permettent de produire des résumés de qualité mais sont très limitées pour traiter les textes longs. Cependant, depuis l'émergence des modèles *transformers* pré-entraînés (Devlin *et al.*, 2018; Vaswani *et al.*, 2017) et les architectures séquence à séquence, les résumés abstractifs produits ont connu un progrès significatif en termes de fluidité et de lisibilité. Parmi les architectures séquence à séquence, on retrouve les modèles à base de mécanismes d'attention (Luong *et al.*, 2015) qui pondèrent les passages selon leur importance dans le document source, les réseaux pointeurs (*pointer networks*) (Vinyals *et al.*, 2015; See *et al.*, 2017) qui commencent par extraire les passages du document les plus importants et procèdent à une reformulation du reste du document par la suite, ou encore les modèles de langues pré-entraînés sur la tâche de résumé abstractif tels que Pegasus (Zhang *et al.*, 2020a) ou T5 (Raffel *et al.*, 2020).

Même si ces derniers ont montré des résultats prometteurs, la tâche reste difficile, notamment en ce qui concerne la génération de résumé dans les domaines de spécialité tels que le juridique, le médical ou les sciences (El-Kassas *et al.*, 2021). En effet, ces domaines exigent des connaissances et une terminologie spécifiques qui ne sont pas toujours bien représentées dans les modèles de langues, car ces derniers ont été pré-entraînés sur des corpus de données génériques (souvent des corpus d'actualité). Par conséquent, la production de résumés de bonne qualité dans ces domaines de spécialité, demeure encore aujourd'hui un défi majeur, notamment lorsqu'il s'agit d'autres langues, telles que le français par exemple (Zhou *et al.*, 2022).

2.2 Évaluation automatique des résumés

Pour évaluer les résumés produits par les systèmes, l'évaluation automatique la plus courante se base sur des résumés de référence (aussi appelés « Gold Standard »), généralement produits de façon manuelle. L'idée est que plus les résumés générés sont proches des résumés de référence, meilleurs ils

sont. Il s'agit donc ensuite de calculer de façon automatique une similarité entre les résumés produits et les résumés de référence.

Dans le cadre du résumé abstractif, deux grandes familles de métriques sont traditionnellement reportées (El-Kassas *et al.*, 2021; Ermakova *et al.*, 2019) :

— Les mesures ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) (Lin, 2004) se basent sur le chevauchement des mots (unigrammes, bigrammes et n-grammes) entre les deux textes à comparer. Elles sont orientées rappel : plus le résumé à évaluer contient de n-grammes du résumé de référence, meilleur est le résultat des métriques.

— Les mesures BLEU (*BiLingual Evaluation Understudy*) (Papineni *et al.*, 2002), initialement créées pour l'évaluation de la traduction automatique de texte, se basent également sur le nombre de n-grammes en commun entre les résumés à évaluer et le résumé de référence. Elles mesurent combien de n-grammes du résumé à évaluer apparaissent dans le résumé de référence, et sont donc pour leur part orientées précision.

Ces métriques, bien que simples à mettre en place, sont incapables de considérer des synonymes ou expressions sémantiquement proches. Il est donc habituel de reporter, en complément des métriques ROUGE et BLEU, des métriques basées sur la similarité sémantique des résumés. Parmi elles, nous pouvons citer BertScore (Zhang *et al.*, 2020b) ou la métrique Cos Embed utilisée dans (Dusart *et al.*, 2023), qui calculent une similarité cosinus entre des représentations générées respectivement par BERT ou Word2Vec.

L'utilisateur intéressé pourra se référer à (Ermakova *et al.*, 2019) pour la description de mesures d'évaluations complémentaires, toutes ayant leurs points forts et faibles. On relèvera cependant deux limitations principales à ces métriques :

— Aucune ne pénalise de façon spécifique la présence d'informations incohérentes (au sens où elles ne sont pas présentes dans le documents d'origine), nommées hallucinations par (Maynez *et al.*, 2020). Or, d'après (Cao *et al.*, 2018), 30 % des résumés produits par les méthodes d'état de l'art contiennent ce genre d'incohérences (*fact fabrication*). Des métriques spécifiques ont été proposées dans l'état de l'art, basées sur l'inférence, dans le cadre des systèmes de question-réponses ou encore les systèmes d'extraction d'information (Ji *et al.*, 2022).

— Aucune ne prend en compte de façon explicite les spécificités métiers des documents considérés. La mise en correspondance de n-grammes ou le calcul de similarités sémantiques peut ne pas suffire à interpréter correctement les résultats.

Afin d'analyser plus finement les résultats, nous proposons dans cet article d'utiliser et discuter des métriques complémentaires basées sur les entités d'intérêt, en évaluant leur couverture mais aussi, sur cette base, les hallucinations produites par les méthodes.

2.3 Collections de test dans le domaine juridique

Le domaine juridique est un domaine en constante évolution, centre d'intérêt croissant pour la communauté du Traitement Automatique des Langues. Cette tendance est mise en évidence par des initiatives telles que la campagne d'évaluation SemEval 2023 [1]. Cette édition propose une tâche couvrant diverses applications du TAL telles que l'analyse du discours et la détection d'entités nommées. De façon complémentaire, plusieurs corpus de textes juridiques sont désormais en libre accès, notamment le corpus belge de recherche d'articles statutaires BSARD (Louis & Spanakis, 2022).

1. https://semeval.github.io/SemEval2023/tasks.html

En ce qui concerne la génération automatique de résumés, bien qu'il existe plusieurs corpus génériques de résumés d'actualité tels que CNN/DM (Nallapati *et al.*, 2016), NYT (Sandhaus, 2008) ou OrangeSum en français (Eddine *et al.*, 2020), ces derniers ne portent pas sur des textes juridiques. Parallèlement, des jeux de données juridiques français pour le résumé ont été développés, tels que le corpus CASS de comptes rendus de la cour de cassation (Bouscarrat *et al.*, 2019), ou encore récemment le corpus EUR-Lex-Sum (Aumiller *et al.*, 2022), qui est un corpus d'articles juridiques provenant de la plateforme de loi de l'Union Européenne. Cependant, à notre connaissance, il n'existe à ce jour aucun corpus en français combinant des problématiques liées à l'actualité et à l'utilisation de vocabulaire métier juridique.

3 Une collection de test pour le résumé de textes juridiques en français

Afin d'évaluer les performances des modèles de génération de résumé abstractif, nous avons identifié une collection de documents de l'actualité juridique française [2]. Cette collection sera notre corpus de travail pour cet article.

La collection est constituée de 8 485 documents de veille juridique et réglementaire pour les collectivités territoriales et les administrations publiques. Chaque document comporte (a) un titre, (b) un texte (corps/contenu), (c) un ensemble de méta-données associées, notamment la thématique du document, et (d) un résumé produit manuellement. Un exemple de document de la collection est présenté dans le tableau 1. Toutes ces informations sont rédigées par des spécialistes du domaine dont l'objectif est de maintenir l'actualité juridique à jour. Ces spécialistes se focalisent sur la nouveauté et l'évolution dans les lois, en commençant souvent par la présentation du contexte de la loi, le code et les différents aléas de cette dernière, suivis de la présentation du changement qui a eu lieu. Chaque résumé reprend le contexte général du contenu du document et l'actualité décrite dans ce dernier.

Si on considère notre tâche de résumé automatique, nous pouvons utiliser cette collection en considérant le contenu de chaque document comme le *document source* et chaque résumé produit manuellement par un éditeur spécialiste en la matière comme le *résumé de référence* (Gold Standard). L'objectif des systèmes que nous évaluons sera donc de produire pour chaque document source un résumé (*résumé généré*) le plus proche possible du résumé de référence. En moyenne, les documents sources et les résumés de référence comportent respectivement 485 et 81 mots. Aucune différence notable n'est observée entre les différentes thématiques.

La collection aborde sept thématiques juridiques. Chaque document est associé à une seule thématique, attribuée par les spécialistes. Le tableau 2 décrit brièvement chacune d'elles en illustrant avec un exemple de titre d'article, et fournit la répartition des documents selon ces différentes thématiques.

2. Cette collection, non diffusable à ce stade, est la propriété de notre partenaire industriel.

Titre : La Cour de cassation communique.
Contenu : Comme le précise sa préface, élaborée en collaboration avec le service de documentation, des études et du rapport et avec le service de communication, cet état des lieux doit paraître chaque mois (à l'exception des mois de juillet et août). Son objectif est de faire connaître l'activité de la chambre criminelle de la Cour de cassation à un public plus large que celui des magistrats, des avocats et des professeurs de droit. Car « la Cour de cassation tranche, en particulier dans le domaine pénal, des questions diverses et difficiles qui, par l'enjeu qui s'y attache, intéressent l'ensemble des citoyens ». Accessible sur le site de la Cour de cassation, elle est envoyée gratuitement par voie électronique à toute personne qui en fait la demande. Les arrêts sont classés par thématiques, par exemple pour ce premier numéro, Audience Blanchiment, Détention provisoire, etc.
Thématique : Justice
Résumé : La Cour de cassation vient de publier le premier numéro d'une sélection commentée de ses arrêts rendus par la chambre criminelle (no 1, juin 2020).

TABLE 1 – Exemple de document juridique avec ses titre, contenu, thématique et résumé.

4 Expérimentations et résultats

Nous présentons dans cette section le protocole d'expérimentations que nous avons utilisé, les modèles que nous avons choisis pour la génération automatique du résumé abstractif ainsi que les résultats obtenus. Il est à noter que les expérimentations que nous avons effectuées n'avaient pas pour objectif d'améliorer les performances des modèles utilisés en termes de génération de résumé, mais plutôt d'évaluer leur capacité de compréhension des textes de nature juridique.

4.1 Protocole expérimental

Pour notre étude, nous avons sélectionné des modèles de génération de résumé abstractif pour lesquels des variantes pré-entraînées en français étaient disponibles. Nous avons récupéré l'ensemble de ces modèles à partir de la plateforme *HuggingFace* [3]. Ces modèles sont tous basés sur une architecture de séquence à séquence, qui se compose d'une partie encodeur qui reçoit le texte source du document en entrée et produit une représentation vectorielle. Cette représentation est ensuite transmise au décodeur, qui a pour rôle de générer le texte du résumé en sortie.

3. https://huggingface.co/

Thématique	Description	Nombre de documents
Commande Publique	Couvre des sujets tels que les règles de passation des marchés publics, la réglementation des délais de paiement, les évolutions récentes du droit de la commande publique, etc. **Exemple :** *« Accord-cadre de travaux à bons de commande : quid du règlement des prestations ? »*.	3 398
Comptabilité et Finances locales	Aborde des sujets tels que la réglementation des budgets des collectivités territoriales, la gestion des dépenses et des recettes, les évolutions récentes du droit des finances locales, etc. **Exemple :** *« Sur quelles perspectives économiques préparer son budget 2023 ? »*.	577
Élections et Démocratie participative	Concerne les lois, règlements et jurisprudences relatifs aux élections en France (élections nationales, régionales, départementales, municipales). **Exemple :** *« Quel est l'office du juge lorsqu'il est saisi d'un compte de campagne ? »*.	303
État civil et Cimetières	Couvre des sujets tels que la tenue des registres d'état civil (naissance, mariage, décès), la délivrance des actes d'état civil, les règles relatives à la gestion et à l'entretien des cimetières, etc. **Exemple :** *« Quid du retour de la France au sein de la Commission internationale de l'état civil ? La réponse est non ! »*.	1 314
Justice	Concerne les lois, règlements et jurisprudences relatifs au système judiciaire français, ainsi qu'aux droits et obligations des acteurs de la justice, tels que les magistrats, les avocats, les victimes, etc. **Exemple :** *« 3 questions sur 10 ans de partenariat entre le Conseil National des Greffiers des Tribunaux de Commerce (CNGTC) et l'Ecole Nationale de la Magistrature (ENM) ! »*.	803
Ressources Humaines territoriales	Aborde des sujets tels que le recrutement, la formation, l'évaluation et la promotion des agents publics, les règles relatives à la discipline et à la sanction des agents, ainsi que les évolutions récentes du droit de la fonction publique territoriale. **Exemple :** *« Il faut remplir les conditions pour requalifier des vacations en CDI »*.	250
Urbanisme	Concerne les lois, règlements et jurisprudences relatifs à l'aménagement du territoire, ainsi que les évolutions récentes du droit de l'urbanisme. **Exemple :** *« Dispositifs de végétalisation de constructions : possibilité de déroger aux règles du PLU »*.	1 840

TABLE 2 – Répartition par thématique des documents juridiques de la collection de test.

4.1.1 Modèles génératifs évalués

Ci-après, nous décrivons les 4 modèles que nous avons utilisés pour le fine-tuning (voir la sous-section suivante 4.1.2 pour les paramètres utilisés) :

– **BART (Lewis *et al.*, 2020)** : il s'agit d'un modèle de langue avec une architecture séquence à

séquence, où l'encodeur est bidirectionnel et le décodeur est auto-régressif.

– BARThez (Eddine *et al.*, 2020) : il s'agit d'un modèle *transformer* français dédié à la génération de résumé, inspiré du modèle BART, possédant 6 couches bidirectionnelles pour l'encodeur ainsi que 6 couches pour le décodeur.

– Bert2Bert (Chen *et al.*, 2022) : il s'agit d'une architecture séquence à séquence ayant comme encoder et decoder une variante du modèle BERT pré-entraîné sur le français.

– T5 (Raffel *et al.*, 2020) : il s'agit d'un modèle de langue ayant une architecture auto-regressive de décodage. T5 a été entraîné sur plusieurs tâches du traitement automatique du langage naturel : la traduction automatique, la réponse aux questions, la génération de résumé, etc.

4.1.2 Fine-tuning des modèles de langue

Comme chacun des modèles utilisés a été pré-entraîné avec un corpus de données générique, généralement issu du web, nous avons décidé de procéder à une étape d'ajustement (*fine-tuning*) des modèles sur les données de spécialité que nous avons utilisées. Les modèles BART et T5 ayant été initialement pré-entrainés sur des corpus anglais, nous avons utilisé des variantes de ces derniers entrainées sur du texte français [4] [5].

La configuration par défaut de chaque modèle a été conservée, nous avons uniquement modifié les hyperparamètres suivants :

— Nombre d'epochs : 10,
— Batch size : 6,
— Nombre de tokens du texte donné en entrée : 512 tokens,
— Nombre de tokens du résumé généré : 100 tokens.

Nous avons utilisé la technique k-*fold* (ou validation croisée) pour le fine-tuning avec $k = 5$. Par conséquent, chaque modèle a été ajusté 5 fois avec à chaque fois une répartition $\frac{4}{5}$ de la collection utilisés pour l'entraînement et $\frac{1}{5}$ de la collection utilisé pour le test. Chaque partie ($\frac{1}{5}$ de la collection) contient 1 697 paires (document source – résumé de référence). Le pourcentage de distribution des thématiques, quant à lui, a été conservé sur la constitution de chacune des parts, c'est-à-dire, à égalité entre chaque part.

4.2 Résultats

Afin d'évaluer les performances des modèles que nous avons ajustés (ou « fine-tunés »), nous avons utilisé deux types distincts de métriques d'évaluation. D'une part, nous avons employé les métriques ROUGE (ROUGE-1, ROUGE-2 et ROUGE-L) (Lin, 2004) et BLEU (Papineni *et al.*, 2002), couramment utilisées dans la tâche de génération de résumés. Ces métriques permettent d'évaluer les résumés produits en termes de correspondance des n-grammes. D'autre part, nous avons utilisé un score de similarité sémantique *CosSim* pour évaluer la correspondance sémantique entre les résumés produits et les résumés de référence. Ce dernier est obtenu en calculant le cosinus entre les représentations des résumés de référence et des résumés générés. Notre collection étant en français, nous avons utilisé le modèle de langue CamemBERT (Martin *et al.*, 2020) pour le calcul

4. airKlizz/bart-large-multi-fr-wiki-news
5. plguillou/t5-base-fr-sum-cnndm

de représentations. Les résultats présentés dans le tableau 3 révèlent que le modèle Bert2Bert se distingue des autres en termes de performances, avec les meilleurs scores ROUGE, BLEU et CosSim. Les modèles BART et BARTHez ont des résultats relativement similaires et, à l'opposé, le modèle T5 montre les plus faibles résultats.

Modèle	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	CosSim
BART	0,34	0,13	0,20	0,29	0,92
BARThez	0,34	0,15	0,22	0,27	0,92
Bert2Bert	**0,39**	**0,19**	**0,25**	**0,36**	**0,93**
T5	0,30	0,08	0,17	0,30	0,89

TABLE 3 – Résultats des différents modèles sur les mesures ROUGE-1, ROUGE-2, ROUGE-L, BLEU (nous reportons la F-mesure) ainsi que le score de similarité sémantique.

Évaluation par thématique

Nous avons ensuite analysé les différents modèles selon chaque thématique de la collection, afin de vérifier si les comportements étaient similaires entre les thématiques. Les résultats de cette analyse sont affichés dans la figure 1 pour la métrique ROUGE-L. Il est à noter que les résultats pour les autres métriques (ROUGE-1, ROUGE-2, BLEU et CosSim) sont comparables et ne permettent pas de dégager de conclusions complémentaires. Ils n'ont donc pas été reportés ici.

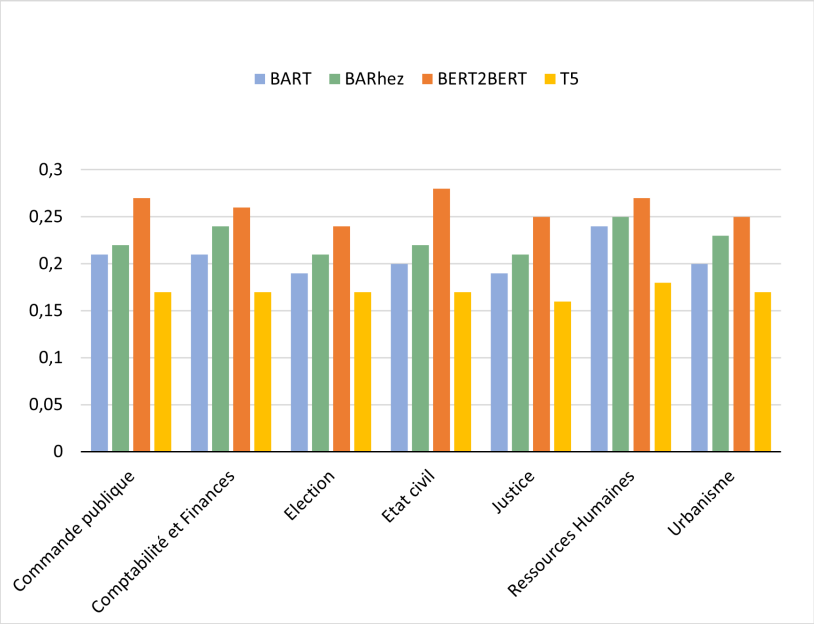


FIGURE 1 – Évaluation des performances des modèles par thématique selon la métrique ROUGE-L.

Nous pouvons distinguer deux observations des résultats de la figure 1. Premièrement, les résultats obtenus pour l'ensemble du corpus sont cohérents avec ceux obtenus pour chacune des thématiques. En effet, le modèle Bert2Bert surpasse les autres modèles sur l'ensemble des domaines, tandis que le modèle T5 obtient les scores les plus bas. Deuxièmement, bien que les documents soient présents en proportion différente dans les thématiques, les scores obtenus d'une thématique à l'autre

sont relativement similaires. Par exemple, le modèle Bert2Bert obtient un score de 0,27 pour les thématiques *Commande Publique* et *Ressouces Humaines territoriales*, bien que leurs occurrences soient très différentes (3398 et 250 respectivement). Ceci peut s'expliquer par le fait que les documents sont rédigés de manière uniforme et similaire, quelle que soit la thématique abordée.

5 Étude détaillée des résultats se basant sur les entités d'intérêt

Les domaines de spécialité tels que le juridique, la santé ou encore les sciences de manière générale sont des domaines particulièrement sensibles, où chaque concept utilisé, chaque nom propre ou même adjectif a une signification concise. Par conséquent, la véracité de l'information véhiculée dans ces domaines tient une place très importante. Les scores ROUGE et BLEU que nous avons obtenus nous permettent d'avoir une première indication sur la qualité des résumés générés par les modèles de langue. Cependant, ils ne nous permettent pas d'évaluer la *couverture* des résumés en terme de vocabulaire d'intérêt (vocabulaire métier et entités nommées), pas plus qu'ils ne permettent d'évaluer les *incohérences* par rapport au document source.

Dans cette section, nous proposons donc de nous focaliser sur une analyse plus poussée des résumés générés. Nous définissons le concept d'*entité d'intérêt*, c'est-à-dire une entité liée au domaine juridique ou une « simple » entité nommée. Dans la suite, nous choisissons d'évaluer la couverture des résumés et leurs incohérences à partir des entités d'intérêt détectées dans le document source et les résumés générés.

Les incohérences des documents sont appelées *hallucinations* dans la littérature. Une hallucination est définie comme une information qui ne peut pas être déduite à partir du document source (Maynez *et al.*, 2020). On distingue deux types d'hallucinations : les hallucinations *intrinsèques* qui correspondent à des mots ou groupes de mots présents dans le document source mais mal utilisés dans le résumé généré (un code de loi mentionné à mauvais escient par exemple), alors que les hallucinations *extrinsèques* correspondent à des mots ou groupes de mots non présents dans le document source. À ce stade, il est important de noter que les hallucinations peuvent être correctes (*factuelles*) : elles peuvent se baser sur une connaissance générale acquise en dehors du document. Les hallucinations intrinsèques étant compliquées à identifier, nous nous focalisons dans cet article sur les hallucinations extrinsèques afin d'évaluer les incohérences.

Une illustration des hallucinations est proposée dans l'exemple du tableau 4. L'exemple donné concerne un article juridique sur l'attribution des prénoms aux enfants à la naissance, qui énumère les différentes lois sur le sujet par ordre chronologique et explique les particularités et nouveautés de chacune d'elles. Dans le tableau, les entités d'intérêt sont en gras. Elles sont en rouge lorsqu'elles correspondent à des hallucinations extrinsèques. Cette notion d'hallucination extrinsèque a son pendant dans le résumé de référence. Les experts se sont en effet ici basés sur leur connaissances préalables et ont introduit une entité que nous considérons comme *abstraction* (en violet).

Dans la suite de cette section, nous détaillons la typologie des entités d'intérêt considérées, puis évaluons nos résultats en termes de couverture et d'hallucination.

Document source : Le choix des prénoms, liberté totale... ou presque. La **loi du 1 avril 1803** avait fixé les règles concernant le choix des prénoms, alors limité aux seuls en usage dans les différents calendriers ainsi qu'à ceux des personnages notoirement illustres dans l'Histoire. Ainsi, l'officier d'état civil n'avait alors qu'un simple rôle de vérification sans autre choix que de refuser tout prénom non conforme à cette prescription légale. Toutefois, la **Cour de cassation** avait admis dès 1981 qu'il n'y avait pas lieu d'exiger que le calendrier invoqué émane d'une autorité officielle. Poursuivant dans cette voie, la **loi n° 93-22 du 8 janvier 1993** a profondément modifié cette architecture en disposant, par l'**article 57 du Code civil**, que « [...]. L'officier d'état civil porte immédiatement dans l'acte de naissance les prénoms choisis. [...] Lorsque ces prénoms ou l'un d'eux, seul ou associé aux autres prénoms ou au nom, lui paraissent contraires à l'intérêt de l'enfant, l'officier de l'état civil en avise sans délai le procureur de la République. Celui-ci peut saisir le juge aux affaires familiales ». Ainsi, d'un système rigide instauré dès la période révolutionnaire, cette loi donne une liberté de principe et quasi-totale aux parents. Désormais, nul besoin de choisir un prénom existant, bien au contraire : il suffit de l'inventer... ou alors de donner à un prénom traditionnel une orthographe moins conventionnelle... L'unique limite à cette liberté est l'intérêt de l'enfant : le choix des prénoms ou leur association ne doit pas lui nuire. La loi donne alors à l'officier d'état civil un rôle central puisque c'est lui seul qui doit déterminer s'il convient ou non d'alerter le **Parquet**.	**Résumé de référence :** Jusqu'en 1993, la **loi du 1er avril 1803** encadrait strictement le choix des prénoms de l'enfant et cantonnait donc les parents au strict choix de prénoms déjà existants. Également, le principe d'immutabilité de l'état des personnes a longtemps limité drastiquement les cas de modification ou de changement de ces mêmes prénoms. C'est ainsi qu'une liberté nouvelle est apparue avec la loi du 8 janvier 1993 dans le domaine du choix et la **loi no 2016-1547 du 18 novembre 2016**, dite « Justice du XXIe siècle », élargit encore cette liberté à la sphère de la modification et du changement de prénom.
	Résumé généré par BART : La **loi n° 93-22 du 8 janvier 1993** portant réforme de la procédure à l'état civil a modifié les règles concernant le choix des prénoms des enfants par les parents. Les parents ne sont plus obligés de choisir un prénom existant pour leur enfant. La **Cour de cassation** s'est prononcée quant au calendrier en affirmant qu'il n'y avait pas lieu d'exiger que ce dernier émane d'une autorité officielle.
	Résumé généré par BARThez : La **loi n 93-22 du 8 janvier 1993** a profondément modifié cette architecture en disposant, par l'**article 71 du Code civil**, que « les prénoms de l'enfant sont choisis par ses père et mère .... L'officier d'état civil porte immédiatement dans l'acte de naissance les prénoms choisis par les parents ou l'un d'eux, seul ou associé aux autres prénoms ou au nom ».
	Résumé généré par Bert2Bert : La **loi n° 93-22 du 18 janvier 1992** a introduit dans le Code civil une disposition spécifique relative à l'attribution du nom de famille à un enfant. En effet, à partir de cette date, les parents ont la quasi-liberté de choisir les prénoms qu'ils désirent attribuer à leurs enfants. L'officier d'état civil est le seul acteur à pouvoir révoquer cette liberté donnée aux parents s'il juge que le prénom choisi nuit à leur enfant.
	Résumé généré par T5 : L'**article 57 du Code civil** prévoit que « le choix des prénoms est effectué par les parents sous le contrôle de l'officier d'état civil dans les actes de naissance, sans distinction de nom ou de prénom. Il n'y a donc pas lieu d'exiger que le choix des prénoms soit associé aux autres prénoms ou au nom de l'enfant choisi [...] ».

TABLE 4 – Exemple d'un document source, du résumé de référence associé et des résumés générés par les différentes approches. Les entités d'intérêt sont repérées en gras. Dans les résumés générés, les hallucinations extrinsèques sont en rouge. L'abstraction du résumé de référence est en violet.

5.1 Typologie des entités d'intérêt considérées

Afin d'identifier les différentes entités d'intérêt présentes dans notre collection, nous avons effectué une analyse manuelle de plusieurs échantillons. Nous avons observé une forte présence d'entités nommées de type personne, organisation ou localisation ainsi qu'un ensemble d'entités liées au domaine juridique. Nous avons extrait les entités nommées à l'aide d'un modèle CamemBERT (Martin *et al.*, 2020). Les entités juridiques quant à elles ont été extraites avec des expressions régulières en ne considérant que les types d'entités juridiques les plus courants et les patrons syntaxiques les plus stables afin d'éviter des biais de résultats. De façon détaillée :

- Nous considérons 4 types d'entités juridiques :
  - Loi
  - Article de loi
  - Proposition de loi
  - Décret
- Ces entités peuvent être exprimées selon 3 patrons syntaxiques différents :
  - Entité N° Numéro : ce patron syntaxique correspond à une entité nommée suivie d'un « N° » (abréviation de « numéro ») et d'un numéro identifiant l'entité en question. Exemple : « Loi n° 2016 - 1547 ».
  - Entité Code - Numéro : ce patron syntaxique correspond à une entité nommée suivie d'un code identifiant l'entité, suivi d'un tiret et d'un numéro. Exemple : « Article L. 2122-18 ».
  - Entité du Date : ce patron syntaxique correspond à une entité nommée suivie de l'article « du » suivi d'une date. Exemple : « Décret du 2 juin 2021 ».

Il convient de noter que ces patrons syntaxiques peuvent parfois être combinés dans une même expression. Par exemple, on peut rencontrer des expressions telles que « Loi n°5125 du 21 janvier 2023 » qui utilisent à la fois les patrons 1 et 3. Par ailleurs, nous avons choisi de ne considérer les dates que lorsqu'elles sont associées aux entités, car leur forme est très variable, et nous n'avons pas pris en compte les entités liées au temps (horaires, périodes de la journée, etc). Enfin, dans le cas où une entité est identifiée à la fois comme entité nommée et comme entité juridique, nous la considérerons exclusivement comme faisant partie de l'ensemble des entités juridiques. En effet, nous avons observé que certaines lois portent des noms de personnes ou de lieux.

5.2 Métriques considérées

Sachant les entités d'intérêt présentées dans la section précédente, nous proposons d'évaluer deux types de métriques : la couverture et le taux d'hallucination / abstraction.

Soient $\mathcal{N}(d), \mathcal{N}(r), \mathcal{N}(g)$, les nombres d'entités d'intérêt présentes respectivement dans le document source d , dans le résumé de référence r (gold standard) et dans le résumé généré g.

Une première catégorie de métrique concerne la couverture des résumés :

- le taux de couverture c_g des résumés générés :

$$c_g = \frac{\mathcal{N}(g \cap d)}{\mathcal{N}(d)} \tag{1}$$

où $\mathcal{N}(g \cap d)$ est le nombre d'entités de d trouvées dans le résumé généré.

— le taux de couverture c_r des résumés de référence :

$$c_r = \frac{\mathcal{N}(r \cap d)}{\mathcal{N}(d)} \tag{2}$$

où $\mathcal{N}(r \cap d)$ est le nombre d'entités de d trouvées dans le résumé de référence rédigé par les experts. c_r peut être vu comme un maximum atteignable par les différents modèles.

Les métriques de couverture se basent toutes sur le document source à résumer ($\mathcal{N}(d)$). Elles diffèrent en cela des métriques proposées par (Nan *et al.*, 2021) qui comparent les résumés générés aux résumés de référence ($\mathcal{N}(r)$). Nous n'avons pas fait ce choix afin d'avoir des métriques généralisables dans le cas où les résumés de référence n'existeraient pas. D'autre part, comparer les entités à celles du document source permet non seulement d'évaluer les résumés générés, mais également les résumés de référence, ce qui n'est à notre connaissance pas fait dans la littérature.

Une deuxième catégorie de métrique est liée à l'apparition d'entités dans les résumés générés/de références, entités qui n'étaient pas présentes dans les documents sources. Nous définissons :

— le taux d'hallucination (extrinsèque) h :

$$h = \frac{\mathcal{N}(\neg g)}{\mathcal{N}(g)} \tag{3}$$

où $\mathcal{N}(\neg g)$ est le nombre d'entités hallucinées dans g. Le taux traduit le pourcentage d'entités hallucinées dans le résumé généré.

— le taux d'abstraction a :

$$a = \frac{\mathcal{N}(\neg r)}{\mathcal{N}(r)} \tag{4}$$

où $\mathcal{N}(\neg r)$ est le nombre d'entités abstraites dans r, c'est-à-dire, le pourcentage d'entités de r qui ne font pas partie des entités de d. Ces abstractions peuvent être comparées aux hallucinations extrinsèques des résumés générés dans le sens où elles ne portent pas sur des connaissances présentes dans d. Elles proviennent des experts qui ont réalisés les résumés de référence : ces derniers peuvent en effet se servir de leurs connaissances *a priori* pour rédiger les résumés. Il est cependant à noter que mêmes si comparables à des hallucinations extrinsèques, les abstractions sont factuelles, c'est-à-dire qu'on peut les considérer comme vraies, au contraire de certaines hallucinations extrinsèques.

À ces métriques, afin d'avoir une vision plus globale, nous ajoutons la proportion de résumés générés touchés par des hallucinations :

$$p_h = \frac{|G_h|}{|G|} \tag{5}$$

et la proportion de résumés de référence concernés par des abstractions :

$$p_a = \frac{|R_a|}{|R|} \tag{6}$$

où G est l'ensemble des résumés générés g, $G_h \in G$ est l'ensemble des résumés générés g contenant au moins une hallucination extrinsèque, R est l'ensemble des résumés de référence et R_a l'ensemble des résumés de référence contenant au moins une abstraction.

5.3 Résultats

Avant d'examiner les résultats en termes de couverture et hallucination, nous avons étudié la répartition et le nombre des entités d'intérêt dans les documents sources et les différents résumés. Cette analyse est illustrée dans la figure 2. Une première observation est que les entités juridiques sont, de façon non surprenante, de loin les plus présentes, à la fois dans les documents source et dans les résumés. Concernant les modèles, T5 se comporte très différemment des autres modèles : il est capable de générer beaucoup plus d'entités (et même beaucoup plus d'entités que le résumé de référence). Nous constatons enfin que les modèles ont d'une manière générale du mal à générer les entités de type personne (tous les modèles ont un nombre d'entités générées inférieur à celui du résumé de référence).

(a) (b)

FIGURE 2 – Comparaison du nombre d'entités présentes : (a) dans les documents sources, (b) dans les résumés de référence et résumés générés par type d'entité.

Le tableau 5 présente les résultats de couverture et hallucination des résumés de référence. Nous pouvons constater que les résumés de référence couvrent faiblement les entités présentes dans le document source (couverture à 0,09), ce qui n'est pas surprenant étant donnée la taille des résumés de référence. D'autre part, 28% des entités considérées sont abstraites. Ces résultats ne remettent cependant pas en question la factualité (véracité) des résumés, étant donné qu'ils ont été rédigés par des spécialistes du domaine.

	Couverture c_r	Abstraction a	Proportion p_a
Résumé de référence	0,09	0,28	0,36

TABLE 5 – Taux de couverture, d'abstraction et proportion des résumés concernés par les abstractions des résumés de référence.

Concernant les résumés générés automatiquement, les résultats du tableau 6 montrent des différences significatives dans les taux obtenus par les différents modèles. Bien que le modèle Bert2Bert ait obtenu les meilleurs scores ROUGE et BLEU, il obtient le taux de couverture le plus faible, le taux d'hallucination de loin le plus élevé (>60%) ainsi que la proportion de résumés hallucinés la plus élevée. Le modèle T5 présente quant à lui la meilleure couverture des entités du document source,

surpassant même celle des résumés de référence. Enfin, le modèle BART présente les taux les plus bas d'hallucination et de proportion d'hallucination.

Modèle	Couverture c_g	Hallucination h	Proportion p_h
BART	0,13	**0,19**	**0,25**
BARThez	0,09	0,21	0,29
Bert2Bert	0,03	0,61	0,72
T5	**0,18**	0,22	0,47

TABLE 6 – Taux de couverture, d'hallucination et proportion des résumés concernés par les hallucinations des résumés générés par les différents modèles.

Nous avons également regardé les résultats sous l'angle de la thématique abordée par les documents. Comme dans la section 4.2, nous n'avons pas observé de différence entre les différentes thématiques et ne reportons donc pas les résultats ici.

Une analyse complémentaire a concerné l'étude des taux de couverture et d'hallucination en fonction des entités concernées (personne, organisation, localisation et juridique). Les hallucinations concernent principalement les entités juridiques, probablement en raison de leur forte présence dans la collection de données. Les entités de type personne, organisation et localisation sont hallucinées de manière relativement similaire.

Enfin, afin d'examiner plus en détail les hallucinations, nous avons calculé un pourcentage d'intersection entre les entités hallucinées des résumés générés et les entités abstraites des résumés de référence. Les résultats sont présentés dans le tableau 7. Ils donnent une indication sur la factualité des hallucinations. Une fois encore, Bert2Bert obtient les résultats les moins convaincants, en contradiction avec les résultats des métriques traditionnelles. Ces analyses doivent cependant être poussées : sans remise en contexte des entités hallucinées, on ne peut pas déduire leur exacte factualité. Elles peuvent en effet être utilisées en provoquant des contre-sens ou de façon erronée.

BART	BARThez	Bert2Bert	T5
30%	22%	20%	27%

TABLE 7 – Pourcentage d'entités hallucinées faisant partie des entités abstraites

Tous ces résultats confirment dans leur ensemble qu'une simple analyse sur les métriques ROUGE et BLEU n'est pas suffisante dans un contexte métier. Le modèle Bert2Bert qui semblait être le plus performant sur les métriques classiques s'avère être celui qui génère le plus d'hallucinations "non contrôlées". Par conséquent, nous envisageons de poursuivre une étude plus détaillée des modèles T5 et Bart.

6 Conclusion

Dans cet article, nous avons identifié une collection d'articles d'actualité juridique. Nous avons ajusté (*fine-tuné*) sur cette collection 4 modèles de langue pour le résumé abstractif. Nous les avons évalué avec les métriques classiques du résumé automatique. Nous avons également mené une analyse détaillée des résumés générés à l'aide de la détection des entités nommées et des entités du domaine juridique. Cette analyse a montré que les scores ROUGE et BLEU ne sont pas suffisants pour évaluer des résumés abstractifs métiers. Cette observation souligne l'importance de prendre en considération des critères supplémentaires d'évaluation des résumés tels que la pertinence et la fidélité des informations produites, qui sont cruciaux dans les domaines de spécialité, tels que le juridique.

Cette étude ouvre sur plusieurs perspectives. À court terme, nous souhaitons poursuivre notre évaluation des hallucinations : (i) en détectant les hallucinations intrinsèques, (ii) en analysant la factualité des hallucinations dans leur ensemble, et (iii) en ajoutant d'autres métriques dédiées à l'évaluation des résumés, telles que les métriques basées sur les modèles questions-réponses (comme QuestEval (Scialom *et al.*, 2021) ou QAGS (Wang *et al.*, 2020)), les métriques basées sur la détection des faits (comme FactCC (Goodrich *et al.*, 2019)) ainsi que les métriques basées sur l'implication textuelle (comme PARENT (Dhingra *et al.*, 2019)).

A plus long terme, les modèles de générations peuvent être améliorés selon deux axes : (i) la limitation des hallucinations, dont une piste réside dans la suppression des abstractions dans les résumés de référence (Nan *et al.*, 2021), et (ii) le contrôle de ces dernières, en apprenant aux modèles à halluciner des informations factuelles (véridiques). Ces deux perspectives pourraient permettre d'obtenir des résultats plus précis et fiables dans la génération de résumés dans le domaine juridique, domaine métier dans lequel la véracité de l'information est cruciale.

Références

AKANI E., FAVRE B. & BECHET F. (2022). Abstraction ou hallucination ? état des lieux et évaluation du risque pour les modèles de génération de résumés automatiques de type séquence-à-séquence. In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale / Travaux originaux*, p. 1–10, Avignon, France : Association pour le Traitement Automatique des Langues.

AUMILLER D., CHOUHAN A. & GERTZ M. (2022). Eur-lex-sum : A multi-and cross-lingual dataset for long-form summarization in the legal domain. *arXiv preprint arXiv :2210.13448*.

BOUSCARRAT L., BONNEFOY A., PEEL T. & PEREIRA C. (2019). Strass : A light and effective method for extractive summarization based on sentence embeddings. *arXiv preprint arXiv :1907.07323*.

CAO Z., WEI F., LI W. & LI S. (2018). Faithful to the original : Fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18 : AAAI Press.

CHEN C., YIN Y., SHANG L., JIANG X., QIN Y., WANG F., WANG Z., CHEN X., LIU Z. & LIU Q. (2022). bert2BERT : Towards reusable pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2134–2148, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.151.

DERNONCOURT F., GHASSEMI M. & CHANG W. (2018). A repository of corpora for summarization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan : European Language Resources Association (ELRA).

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*.

DHINGRA B., FARUQUI M., PARIKH A., CHANG M.-W., DAS D. & COHEN W. W. (2019). Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv :1906.01081*.

DOU Z., LIU P., HAYASHI H., JIANG Z. & NEUBIG G. (2021). Gsum : A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2021*, p. 4830–4842 : Association for Computational Linguistics.

DUSART A., PINEL-SAUVAGNAT K. & HUBERT G. (2023). Tssubert : How to sum up multiple years of reading in a few tweets. *ACM Trans. Inf. Syst.* DOI : 10.1145/3581786.

EDDINE M. K., TIXIER A. J.-P. & VAZIRGIANNIS M. (2020). Barthez : a skilled pretrained french sequence-to-sequence model. *arXiv preprint arXiv :2010.12321*.

EL-KASSAS W. S., SALAMA C. R., RAFEA A. A. & MOHAMED H. K. (2021). Automatic text summarization : A comprehensive survey. *Expert Systems with Applications*, **165**, 113679.

ERMAKOVA L., COSSU J. & MOTHE J. (2019). A survey on evaluation of summarization methods. *Inf. Process. Manag.*, **56**(5), 1794–1814.

FABBRI A., LI I., SHE T., LI S. & RADEV D. (2019). Multi-news : A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 1074–1084 : Association for Computational Linguistics.

GOODRICH B., RAO V., LIU P. J. & SALEH M. (2019). Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 166–175.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

HUANG Z., XU W. & YU K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv :1508.01991*.

JI Z., LEE N., FRIESKE R., YU T., SU D., XU Y., ISHII E., BANG Y., MADOTTO A. & FUNG P. (2022). Survey of hallucination in natural language generation. *ACM Comput. Surv.* Just Accepted, DOI : 10.1145/3571730.

LEWIS M., LIU Y., GOYAL N., GHAZVININEJAD M., MOHAMED A., LEVY O., STOYANOV V. & ZETTLEMOYER L. (2020). Bart : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, p. 7871–7880.

LIN C.-Y. (2004). ROUGE : A package for automatic evaluation of summaries. *Text Summarization Branches Out*, p. 74–81.

LOUIS A. & SPANAKIS G. (2022). A statutory article retrieval dataset in French. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 6789–6803, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.468.

LUONG T., PHAM H. & MANNING C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1412–1421, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1166.

MA C., ZHANG W. E., GUO M., WANG H. & SHENG Q. Z. (2022). Multi-document summarization via deep learning techniques : A survey. *ACM Comput. Surv.*, **55**(5). DOI : 10.1145/3529754.

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.

MAYNEZ J., NARAYAN S., BOHNET B. & MCDONALD R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 1906–1919, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.173.

NALLAPATI R., ZHOU B., GULCEHRE C., XIANG B. *et al.* (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv :1602.06023*.

NAN F., NALLAPATI R., WANG Z., NOGUEIRA DOS SANTOS C., ZHU H., ZHANG D., MCKEOWN K. & XIANG B. (2021). Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 2727–2733, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.eacl-main.235.

PAPINENI K., ROUKOS S., WARD T. & ZHU W. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, p. 311–318 : ACL.

QI W., YAN Y., GONG Y., LIU D., DUAN N., CHEN J., ZHANG R. & ZHOU M. (2020). Prophetnet : Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : Findings, EMNLP 2020*, p. 2401–2410 : Association for Computational Linguistics.

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.

RUMELHART D. E., HINTON G. E. & WILLIAMS R. J. (1986). Learning representations by back-propagating errors. *nature*, **323**(6088), 533–536.

SAINI N., SAHA S. & BHATTACHARYYA P. (2019). Multiobjective-based approach for microblog summarization. *IEEE Trans. Comput. Soc. Syst.*, **6**(6), 1219–1231.

SANDHAUS E. (2008). The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, **6**(12), e26752.

SCIALOM T., DRAY P.-A., GALLINARI P., LAMPRIER S., PIWOWARSKI B., STAIANO J. & WANG A. (2021). Questeval : Summarization asks for fact-based evaluation. *arXiv preprint arXiv :2103.12693*.

SEE A., LIU P. J. & MANNING C. D. (2017). Get to the point : Summarization with pointer-generator networks. *arXiv preprint arXiv :1704.04368*.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

VINYALS O., FORTUNATO M. & JAITLY N. (2015). Pointer networks. In *NIPS*.

WANG A., CHO K. & LEWIS M. (2020). Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv :2004.04228*.

ZHANG J., ZHAO Y., SALEH M. & LIU P. (2020a). PEGASUS : Pre-training with extracted gap-sentences for abstractive summarization. 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. 11328–11339 : PMLR.

ZHANG T., KISHORE V., WU F., WEINBERGER K. Q. & ARTZI Y. (2020b). Bertscore : Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* : OpenReview.net.

ZHONG M., LIU P., CHEN Y., WANG D., QIU X. & HUANG X. (2020). Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, p. 6197–6208 : Association for Computational Linguistics.

ZHOU Y., PORTET F. & RINGEVAL F. (2022). Effectiveness of french language models on abstractive dialogue summarization task.

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

Augmentation de jeux de données de RI pour la recherche conversationnelle à initiative mixte

Pierre Erbacher[1] Philippe Preux[2] Jian-Yun Nie [3] Laure Soulier [1]
(1) Sorbonne University, ISIR, Paris
(2) INRIA Scool Team, Lille
(3) Montreal University, Canada
pierre.erbacher@isir.upmc.fr

RÉSUMÉ

Une des particularités des systèmes de recherche conversationnelle est qu'ils impliquent des initiatives mixtes telles que des questions de clarification générées par le système. L'évaluation de ces systèmes à grande échelle sur la tâche finale de RI est très difficile et nécessite des jeux de données adéquats contenant de telles interactions. Cependant, les jeux de données actuels se concentrent uniquement sur les tâches traditionnelles de RI ad hoc ou sur les tâches de clarification de la requête. Pour combler cette lacune, nous proposons une méthodologie pour construire automatiquement des jeux de données de RI conversationnelle à grande échelle à partir de jeux de données de RI ad hoc afin de faciliter les explorations sur la RI conversationnelle. Nous effectuons une évaluation approfondie montrant la qualité et la pertinence des interactions générées pour chaque requête initiale. Cet article montre la faisabilité et l'utilité de l'augmentation des jeux de données de RI ad-hoc pour la RI conversationnelle.

ABSTRACT

Augmenting Ad-Hoc IR Dataset for Mixed-initiative Conversational Search

A peculiarity of conversational search systems is that they involve mixed-initiatives such as system-generated query clarifying questions. Evaluating those systems at a large scale on the end task of IR is very challenging, requiring adequate datasets containing such interactions. However, current datasets only focus on either traditional ad-hoc IR tasks or query clarification tasks We propose a methodology to automatically build large-scale conversational IR datasets from ad-hoc IR datasets in order to facilitate explorations on conversational IR. We perform a thorough evaluation showing the quality and the relevance of the generated interactions for each initial query. This paper shows the feasibility and utility of augmenting ad-hoc IR datasets for conversational IR.

MOTS-CLÉS : Recherche conversationnelle, recherche d'informations, interactions mixtes, méthodologie de construction d'ensembles de données..

KEYWORDS: Conversational search, information retrieval, mixed-initiative interactions, dataset building methodology.

1 Introduction

Les systèmes conversationnels, y compris les assistants personnels et les chatbots, sont de plus en plus populaires pour une grande variété de tâches, notamment la recherche d'informations (RI) en ligne. Bien que des modèles de langue (ML) récents, comme ChatGPT d'OpenAI (Ouyang *et al.*, 2022), aient démontré leur capacité à répondre à des questions factuelles, ceux-ci ne peuvent pas

être considérés comme des systèmes de recherche conversationnelle. En effet, ces derniers sont entraînes à générer le texte le plus probable sans faire explicitement référence aux sources et sans garantie de véracité, ce qui amplifie les biais potentiels et les fausses vérités observés dans les données d'apprentissage (Bender *et al.*, 2021). Pour surmonter cette limitation, les systèmes de recherche conversationnelle doivent s'appuyer sur des capacités de recherche d'informations pour localiser les sources/documents pertinents (Shah & Bender, 2022; Dalton *et al.*, 2022; Zamani *et al.*, 2022; Anand *et al.*, 2020; Bender *et al.*, 2021). Pour améliorer la qualité des réponses, des modèles tels que LaMDA (Thoppilan *et al.*, 2022), WebGPT (Glaese *et al.*, 2022) ou Sparrow (Shuster *et al.*, 2022) peuvent additionnellement conditionner la génération de réponses à des informations récupérées par un outil de recherche d'information indépendant. Ceci ne garantit pas que les informations utilisées soit pertinentes, conduisant potentiellement à des réponses non véridiques ou non informatives (Nakano *et al.*, 2021). Comme le suggère (Dalton *et al.*, 2020a) il est important d'inclure les capacités de recherche d'information dans l'évaluation des modèles de recherche conversationnelle dans leur ensemble. Au-delà de la génération de réponses en langage naturel, l'une des principales capacités des systèmes de recherche conversationnelle est leur participation (pro)active à la conversation avec les utilisateurs afin de les aider à clarifier ou à affiner leurs besoins en information (initiative mixte). (Shah & Bender, 2022; Chu-Carroll & Brown, 1997; Dalton *et al.*, 2022; Zamani *et al.*, 2022; Anand *et al.*, 2020; Bender *et al.*, 2021; Radlinski & Craswell, 2017; Trippas *et al.*, 2020; Aliannejadi *et al.*, 2019; Keyvan & Huang, 2022; Zamani *et al.*, 2020b).

Les travaux pour évaluer la RI conversationnelle (Sekulić *et al.*, 2021; Aliannejadi *et al.*, 2019; Salle *et al.*, 2021; Bi *et al.*, 2021) se concentrent principalement sur l'évaluation de la qualité de la génération de questions de clarification à l'aide de jeux de données alignés, tels que Qulac (Aliannejadi *et al.*, 2019) et ClariQ (Aliannejadi *et al.*, 2021) qui contiennent des paires de requêtes et de questions de clarification, mais seulement sur 237 sujets différents. D'autres taches comme TREC CAsT (Dalton *et al.*, 2020b) se concentrent sur l'évaluation des capacités à retrouver les documents pertinents sans prendre en compte les interactions à initiative mixte. Ces limites démontrent la nécessité de construire des ensembles de données de RI à grande échelle contenant non seulement les requêtes de l'utilisateur mais aussi les interactions mixtes, utilisateur-système, mais aussi des signaux de pertinence. La collecte de telles données conversationnelles est un défi, en raison du coût élevé d'annotation.

Dans le contexte des systèmes de recommandation (Li *et al.*, 2018b; Zhou *et al.*, 2020a; Moon *et al.*, 2019; Kang *et al.*, 2019a; Liu *et al.*, 2021; Jia *et al.*, 2022) ont construit des jeux de données conversationnels en utilisant des annotateurs jouant le rôle de l'utilisateur et du système de recommandation autour d'objectifs et d'éléments prédéfinis. Grâce à la génération automatique, d'autres méthodes proposent de simuler les dialogues à partir de données de recommandation existantes (Gao *et al.*, 2022; Kang *et al.*, 2019b; Wu *et al.*, 2020; Zhou *et al.*, 2020b; Fu *et al.*, 2020). Les tentatives ci-dessus ont été possibles car les données de recommandation contiennent des articles annotés avec des catégories limitées et des caractéristiques discrètes telles que la couleur, la marque ou la gamme de prix. En RI, le transfert de ces approches est difficile dans la mesure où ces caractéristiques ne sont pas discrètes ou pas facilement identifiables, et les besoins en information sont beaucoup plus diversifiés. Les jeux de données disponibles sont très limités. Le jeu de données qulac fournit 10000 interactions à tour unique avec un tuple (intention, requête, question de clarification, réponse) sur 200 sujets seulement (Aliannejadi *et al.*, 2019). C'est loin d'être suffisant pour entraîner ou évaluer les techniques de RI conversationnelle dans divers contextes.

Dans notre travail, nous visons à générer automatiquement des interactions à initiatives mixtes entre un utilisateur et un système et proposer une méthodologie pour augmenter les jeux de données de RI ad-hoc. Pour ce faire, nous concevons un générateur de question de clarificationainsi qu'une

simulation utilisateur. Nous les utilisons pour générer des interactions à initiative mixte sur le jeu de données de RI MsMarco.

2 Etat de l'art

2.1 Évaluation de la recherche conversationnelle

L'évaluation des systèmes de RI conversationnelle reste un défi pour la communauté car cela implique d'évaluer la capacité du système à aider et guider l'utilisateur dans sa recherche. (Dalton *et al.*, 2022). Un tel système de recherche doit donc être capable de 1) générer des questions pour clarifier/expliquer les besoins en information des utilisateurs, et 2) récupérer des documents fournissant des informations pertinentes.

En ce qui concerne la tâche de question-réponse (QA), des jeux de données conversationnels (e.g., coQA (Reddy *et al.*, 2019)) ont été construits à partir de données QA "one-shot" telle que Squad (Rajpurkar *et al.*, 2018), Quac (Choi *et al.*, 2018), ELI5 (Fan *et al.*, 2019), ou OpenQA (Chen *et al.*, 2017) en évaluant aussi la capacité à retrouver les contextes pertinents pour répondre aux questions .

Malgré cette démarche intéressante, ces jeux de données sont insuffisants pour la RI car ils se concentrent généralement sur des questions factuelles au lieu de questions complexes ou exploratoires qui caractérisent les besoins en information.Le jeu de données TREC CAsT (Dalton *et al.*, 2020b) étend la portée des questions et aborde différentes facettes d'information au sein de la conversation (une facette peut être considérée comme une sous-catégorie spécifique du sujet). Cependant, le nombre de dialogues disponibles est très limité. D'autres jeux de données, tels que CANARD (Elgohary *et al.*, 2019), se concentrent sur le raffinement ou la reformulation des requêtes, sans interactions proactives de la part du système.

Le jeu de données MIMICS (Zamani *et al.*, 2020a) contient des questions de clarification de domaine ouvert à grande échelle recueillies auprès d'utilisateurs réels sur le moteur de recherche Bing.Cependant, ce jeu de données ne fournit pas de jugements de pertinence des documents ni d'interactions conversationnelles entre l'utilisateur et le système. Les jeux de données Qulac et ClariQ contiennent à la fois des jugements de pertinence de documents et les conversations mixtes associées. Ils sont construits à partir de la collection TREC Web Track 2009-12, qui fournit des paires de sujets et de facettes annotés, associés à des documents pertinents. Les réponses des utilisateurs ont été collectées par le biais de plateformes de crowd-sourcing. Cependant, la collecte de ces interactions a été coûteuse et les jeux de données restent petits avec seulement 237 sujets et 762 facettes de sujets. Cela est limité pour l'entraînement et l'évaluation des systèmes de recherche conversationnelle.

Face au manque de jeux de données adéquats, une idée de plus en plus répandue dans la communauté consiste à s'appuyer sur la simulation d'utilisateurs pour évaluer les systèmes de recherche conversationnelle : (Erbacher *et al.*, 2022; Salle *et al.*, 2021). Les simulations d'utilisateurs, qui consistent à imiter les requêtes et les réponses des utilisateurs, permettent d'évaluer diverses stratégies sans avoir de trajectoires de conversations prédéfinies dans les données. Par exemple, Salle et al (Salle *et al.*, 2021) évaluent leurs systèmes de clarification de requêtes avec une simulation utilisateur visant à générer des réponses. La simulation d'utilisateur est également exploitée pour concevoir des cadres d'évaluation pour les systèmes de recommandation conversationnels (Kang *et al.*, 2019b; Gao *et al.*, 2022; Wu *et al.*, 2020; Zhou *et al.*, 2020b; Fu *et al.*, 2020), donnant lieu à de grandes interactions de dialogue synthétique à partir de jeux de données de recommandation ad hoc. Cependant, dans le contexte de la recommandation, les conversations sont générées grâce a des contraintes de recherche explicites sur des caractéristiques annotées comme la gamme de prix, la couleur, l'emplacement, le

genre de film ou la marque (Asri *et al.*, 2016; Schatzmann *et al.*, 2007; Peng *et al.*, 2018; Li *et al.*, 2017; Kreyssig *et al.*, 2018) Malheureusement, des approches similaires ne peuvent pas être utilisées pour les tâches de recherche complexes et exploratoires (Belkin & Croft, 1992). Dans la RI à domaine ouvert, les facettes qui sous-tendent les besoins en information ne sont pas nécessairement discrètes ou facilement identifiables, ce qui rend beaucoup plus difficile l'identification et l'annotation des besoins des utilisateurs.

2.2 Question de clarification

Poser des questions de clarification est une tâche conversationnelle qui permet à l'utilisateur de participer au processus de désambiguïsation des requêtes en interagissant avec le système. Aliannejadi et al. (Aliannejadi *et al.*, 2019) proposent un classifieur qui sélectionne itérativement une question de clarification à chaque tour de conversation parmi un ensemble de questions prédéfini. Bi et al. (Bi *et al.*, 2021) complètent cette approche avec une détection d'intention basée sur les retours négatifs et un BERT basé sur la pertinence marginale maximale. Cependant, l'utilisation d'un ensemble de questions fixes limite la couverture des sujets, et donc l'efficacité de l'approche. Une deuxième ligne de travaux vise plutôt à générer des questions de clarification. Dans (Salle *et al.*, 2021), Salle et al. utilisent des modèles et des facettes collectés à partir de l'API Autosuggest Bing pour générer des questions de clarification. À chaque tour de la conversation, ils sélectionnent une nouvelle facette pour générer la question jusqu'à ce que la réponse de l'utilisateur soit positive. Sekulić et al (Sekulić *et al.*, 2021) proposent d'améliorer encore la fluidité en utilisant un LLM pour conditionner la génération de questions de clarification à la requête initiale et à une facette. Par exemple, la requête 'Tell me about kiwi', conditionnée aux facettes 'information fruit' ou 'biology birds' peut générer des questions telles que 'Are you interested in kiwi fruit ?' ou 'Are you interested in the biology of kiwi birds ? Ils s'appuient sur le jeu de données Clariq pour affiner GPT2, et ont constaté que les questions générées sont plus naturelles et utiles que les méthodes basées sur des modèles. Ils ont étendu ce travail en générant des questions à l'aide de facettes extraites des documents récupérés (Sekulić *et al.*, 2022). Zamani et al. (Zamani *et al.*, 2020a) proposent de générer des questions de clarification associées à de multiples facettes (panneaux de clarification) qui sont collectées à l'aide de données de reformulation de requêtes, les clics utilisateurs sont aussi collectés. En revanche, ces données sont construites seulement sur les reformulations de requêtes les plus probables et ne sont pas associées une collection ou des signaux de pertinence sur les documents.

Cet état de l'art met en évidence le manque de données à grande échelle adéquats contenant des interactions à initiatives mixtes pour la tâche de RI. Sachant que la collecte de ces jeux de données avec des annotations humaines serait coûteuse, nous pensons qu'une alternative possible est de générer automatiquement des interactions à initiative mixte à partir des collections existantes.

3 Simulation des interactions à initiatives mixtes et génération de jeux de données de RI conversationnelle

3.1 Définition du problème

Nous présentons notre méthodologie pour générer automatiquement des ensembles de données de RI à grande échelle et à initiative mixte. Pour ce faire, nous proposons d'augmenter les ensembles de données de RI ad hoc avec des interactions utilisateur-système simulées, à savoir des questions de clarification (pour le côté système) et les réponses correspondantes (pour le côté utilisateur).

Pour fournir un jeu de données utiles à l'entraînement de modèles neuronaux de recherche d'information avec des questions de clarification et des signaux de pertinence, il est important de fournir un large éventail d'interactions, à savoir des questions de clarification qui donnent lieu à des réponses positives ou négatives. En gardant à l'esprit qu'un sujet peut être complexe ou ambigu, nous suivons les travaux précédents (Sekulić *et al.*, 2021; Zamani *et al.*, 2020a; Salle *et al.*, 2021) en exploitant les facettes pour générer ces questions de clarification. L'extraction de facettes positives ou négatives autour d'un sujet peut être considérée comme un proxy pour limiter la génération de questions de clarification qui attendent des réponses par "oui" ou "non". De plus, pour assurer la qualité des interactions, nous proposons d'introduire une autre variable de contrainte modélisant l'intention de recherche de l'utilisateur. La paire des variables facette et intention permet de générer des questions de clarification positives et négatives (grâce à la facette) en gardant toujours la génération de réponses cohérentes avec les jugements de pertinence dans le jeu de données initial (grâce à l'intention). Autrement dit, l'échantillonnage de différentes paires facette-intention à partir de passages dont le jugement de pertinence est connu permet de constituer un jeu de données avec des interactions mixtes positives et négatives qui reflètent l'intention de recherche de l'utilisateur. Pour des raisons de simplicité, nous ne considérons que les interactions à un seul tour, et discutons de l'extension aux interactions à plusieurs tours dans la section 6.

Considérons un jeu de données de RI ad hoc $\mathcal{D} = \{\mathcal{P}, \mathcal{Q}, \mathcal{R}\}$, dans lequel $\mathcal{P}$ est une collection de passages (ou documents), $\mathcal{Q}$ est un ensemble de requêtes, et $\mathcal{R}$ est un ensemble de jugements de pertinence. L'ensemble $\mathcal{R}$ comprend des tuples $(q, \mathcal{P}_q^+, \mathcal{P}_q^-)$ indiquant les passages pertinents $\mathcal{P}_q^+ \subset \mathcal{P}$ et non pertinents $\mathcal{P}_q^- \subset \mathcal{P}$, pour une requête $q \in \mathcal{Q}$. Nous supposons que $\mathcal{P}_q^- \cap \mathcal{P}_q^+ = \varnothing$. Notre objectif est d'augmenter ce jeu de données $\mathcal{D}$ avec un ensemble d'interactions à initiative mixte $X = \{X_1, \ldots, X_i, \ldots, X_n\}$. Nous notons une interaction à initiative mixte $X_i = (q, cq, a)$ où q désigne une requête initiale, cq une question de clarification et a la réponse associée. Dans cette optique, nous concevons une méthodologie de construction de jeu de données $\mathcal{M} : D \cup \{X_1, \ldots, X_i, \ldots, X_n\}$ reposant sur deux étapes principales : 1) l'extraction des facettes (positives et négatives) f liées à chaque sujet (si elles ne sont pas disponibles dans le jeu de données initial de RI ad-hoc) qui est ensuite utilisé pour contraindre la génération de questions de clarification, et 2) la génération d'interactions à initiative mixte étant donné une requête q et cette facette f. Selon le jeu de données, les ensembles de facettes positives $\mathcal{F}^+$ et négatives $\mathcal{F}^-$ associés à la requête q peuvent être disponibles ou doivent être construits (section 3.2). Nous supposons également que l'intention de recherche int de l'utilisateur est caractérisée par les documents pertinents disponibles dans le jeu de données initial. Nous proposons ensuite de générer une interaction d'initiative mixte X_i étant donné une requête q et les variables de contrainte f et int. Nous nous appuyons sur 1) un modèle de clarification $\mathcal{CM} : q, f \rightarrow cq$ visant à générer une question de clarification cq autour de la facette f étant donné le sujet de la requête q, et 2) une simulation utilisateur $\mathcal{US} : (cq, int, f) \rightarrow a$ qui infère la réponse a en réponse à la question de clarification cq étant donné la pertinence de la facette f et l'intention de l'utilisateur int.
Nous présentons ci-dessous la méthode d'extraction des facettes, ainsi que les composants permettant de générer des questions et des réponses. Des exemples d'interactions attendues sont présentés dans le Tableau 1.

3.2 Extraction de Facettes

Les facettes peuvent être explicites ou implicites selon le jeu de données. Par exemple, elles sont spécifiées dans TREC Web 2009-12 (Clarke *et al.*, 2009), et par conséquent, Qulac et ClariQ (Over, 2001)). Si elles ne sont pas explicitement spécifiées, nous proposons de les extraire des documents. Des travaux antérieurs ont montré que les facettes des requêtes peuvent être extraites des documents

Exemple d'Interactions simulées	
Query : find phone number for irs **System** : are you looking for phone numbers for the irs internal revenue service ? **Answer** : yes **Intent** : Internal Revenue Service Phone Numbers. 800-829-1040 For individual and joint filers who need procedural or tax law information and/ or help to file their 1040-type IRS Tax Help Line for individual returns (including Individuals| Schedules C and E) ; and, general account information for Form 1040 Filers. **Facet** : internal revenue service phone numbers	**Query** : lps laws definition **System** : are you looking for a definition of a limited partnership ? **Answer** : no **Intent** : The Court will not let you establish an LPS conservatorship unless it finds beyond a reasonable doubt, that the mentally ill person, is gravely disabled. Gravely disabled means that, because of a mental disorder, the person cannot take care of his/her basic, personal needs for food, clothing, or shelter. **Facet** : limited partnership business

TABLE 1 – Exemples d'interactions simulées appartenant au jeu de données MiMarco. Dans le premier exemple, l'intention et la facette sont extraites d'un passage pertinent. Dans le deuxième exemple, l'intention est extraite d'un passage pertinent, mais la question de clarification fait référence à une facette négative du sujet.

les mieux classés (Dou *et al.*, 2016; Kong & Allan, 2013). Inspirés par l'analyse fournie par Sekulić et al. (Sekulić *et al.*, 2022), nous extrayons les mots-clés contextuels les plus importants pour représenter les facettes, comme suggéré dans (Sharma & Li, 2019). L'objectif de l'extraction de facettes est de fournir des mots-clés supplémentaires qui peuvent être utilisés pour générer ultérieurement une question de clarification sur divers sujets ou sous-thèmes. Dans ce travail, les facettes sont un ensemble de mots-clés fournissant un contexte supplémentaire à la requête. Nous la formulons comme une fonction bijective $\psi(P) :\rightarrow \mathcal{F}$ qui fait correspondre un ensemble P de passages à un ensemble de facettes. Étant donné une requête q, nous construisons les ensembles $\mathcal{F}^+$ et $\mathcal{F}^-$ de facettes positives et négatives à partir des ensembles de passages respectivement pertinents et non pertinents, respectivement $\mathcal{P}_q^+$ et $\mathcal{P}_q^-$. Cela nous permet de conserver la pertinence des facettes. Pour ce faire, pour un passage $p \in (\mathcal{P}_q^+ \cup \mathcal{P}_q^-)$, nous extrayons comme facette $f \in \mathcal{F}$ l'ensemble des K mots du passage qui sont les plus similaires avec la représentation du passage (c'est-à-dire avec le jeton [CLS]). Pour calculer la similarité, nous utilisons un modèle Sentence-Bert (MiniLM-L6-v2) pré-entraîné (Reimers & Gurevych, 2019) .

3.3 Génération des Interactions

3.3.1 Génération des questions de clarification

L'objectif du modèle de clarification $\mathcal{CM}$ est de poser des questions de clarification pertinentes relatives à une ambiguïté de la requête. Dans la plupart des modèles proposés (Zamani *et al.*, 2020a; Sekulić *et al.*, 2022; Sekulić *et al.*, 2021; Salle *et al.*, 2021; Aliannejadi *et al.*, 2019), cette ambiguïté est traitée en utilisant le concept de facette. Ainsi, la génération de questions de clarification cq est conditionnée par la requête initiale q et une facette f :

$$p(cq|q, f) = \Pi_i p(cq_i|cq_{<i}, q, f) \tag{1}$$

ou q_i est le i^e jeton de la séquence et $q_{<i}$ les jetons décodés.

3.3.2 Simulation Utilisateur

L'objectif de la simulation utilisateur $\mathcal{US}$ est d'imiter la réponse de l'utilisateur en réponse à une question de clarification compte tenu de son intention. La simulation utilisateur doit donner des réponses utiles aux questions pour aider le système à comprendre son intention. L'intention est une représentation du besoin d'information liée à la requête initiale. Elle est utilisée pour orienter la réponse de la simulation utilisateur vers cet objectif. (Kang *et al.*, 2019b; Gao *et al.*, 2022; Wu *et al.*, 2020; Zhou *et al.*, 2020b; Fu *et al.*, 2020; Erbacher *et al.*, 2022). Nous limitons la question de clarification à demander si l'intention porte sur une facette et la réponse de la simulation d'utilisateur à "oui" ou "non". Cette forme limitée de réponse est motivée par deux raisons : (1) malgré la simplicité, une réponse correcte de cette forme correspond aux interactions réalistes de base avec les utilisateurs et est très utile pour que le système puisse mieux identifier l'intention derrière la requête. (2) Cette forme simple de question et de réponse est plus facile à générer et à évaluer. Plus formellement, la simulation de l'utilisateur vise à estimer la probabilité d'une réponse $a \in \{yes, no\}$ étant donné une requête q, une intention de recherche int, et une question de clarification : $p(a|q, int, cq)$.

Intention de recherche.L'intention de l'utilisateur correspond au besoin d'information de l'utilisateur et n'est connue que par ce dernier. Bien que de multiples représentations de l'intention puissent être adoptées (comme une description détaillée du besoin d'information (Aliannejadi *et al.*, 2019, 2021), une représentation vectorielle (Erbacher *et al.*, 2022) ou des contraintes (Kang *et al.*, 2019b; Gao *et al.*, 2022; Wu *et al.*, 2020; Zhou *et al.*, 2020b; Fu *et al.*, 2020)), les jeux de données de RI n'ont généralement pas d'intention annotée associée à la requête. Cependant, les passages pertinents sont connus dans un jeu de données de RI. Dans cet article, nous utilisons un passage pertinent échantillonné $p \in \mathcal{P}_q^+$ et assimilons son contenu à l'intention sous-jacente int. Formellement : $int \leftarrow p$. Nous reconnaissons que ce choix repose sur une hypothèse forte et nous en discutons dans la section 7.

3.4 Utilisation des interactions d'initiative mixte pour adapter les jeux de données RI ad hoc à la RI conversationnelle

Étant donné un jeu de données de RI ad hoc $\mathcal{D}$, notre objectif est d'augmenter $\mathcal{D}$ avec des conversations d'initiative mixte X. Nous distinguons la création des jeux de données d'entraînement et de test, car ils ont des objectifs différents. Les données d'entraînement nécessitent d'inclure des interactions positives et négatives pour permettre d'entraîner correctement les modèles neuronaux orientés RI face à divers scénarios. Pour rappel, ces interactions positives/négatives sont construites sur la base de documents pertinents et non pertinents déterminant des facettes positives et négatives. L'utilisation de la même heuristique pour générer un jeu de données de test n'est pas appropriée car cela impliquerait d'inclure les jugements de pertinence comme sources de preuves de la génération de questions de clarification à l'étape d'inférence. Par conséquent, nous proposons de concevoir une méthodologie d'évaluation en ligne, exploitant le modèle de clarification $\mathcal{CM}$ et la simulation utilisateur $\mathcal{US}$ pour générer des interactions sans introduire de biais lié aux jugements de pertinence. Nous présentons ces deux méthodologies visant à générer des jeux de données hors ligne et en ligne dans ce qui suit.

3.4.1 Construction un jeu d'entraînement hors ligne avec des jugements de pertinence

Notre méthodologie hors ligne vise à générer un large éventail d'interactions positives et négatives sur la base d'un jeu de données RI ad-hoc. Pour ce faire, nous utilisons des documents pertinents et non pertinents pour construire des facettes positives et négatives contraignant la génération de questions de clarification. Comme contrainte de qualité supplémentaire dans la supervision du jeu de données, nous souhaitons nous assurer que les réponses correspondent à la pertinence des documents utilisés.

Autrement dit, la simulation utilisateur présentée dans la section 3.3.2 est remplacée par une simple heuristique faisant correspondre les réponses a avec la pertinence des facettes f (équation 2 page suivante).

Require: $\mathcal{D} = \{\mathcal{P}, \mathcal{Q}, \mathcal{R}\}$
  $X \leftarrow \{\}$ ▷ Ensemble d'interactions RI
  for $q \in \mathcal{Q}$ **do**
    $\mathcal{F}^+ \leftarrow \psi(\mathcal{P}_q^+)$ ▷ Extraction facets positive
    $\mathcal{F}^- \leftarrow \psi(\mathcal{P}_q^-)$ ▷ Extraction facets négative
    for $f \in (\mathcal{F}^+ \cup \mathcal{F}^-)$ **do**
      $cq \leftarrow \mathcal{CM}(q, f)$ ▷ Génération de questions
      if $f \in \mathcal{F}_q^+$ **then** ▷ Construction réponse
        $a \leftarrow$ *'yes'*
      else
        $a \leftarrow$ *'no'*
      end if
      $X_i = (q, cq, a)$
      $X \leftarrow X \uplus X_i$
    end for
  end for
  return $\mathcal{D} \cup X$

Algorithm 1 – Méthodologie pour construire un jeu de donnée d'entraînement à initiative mixé pour la RI

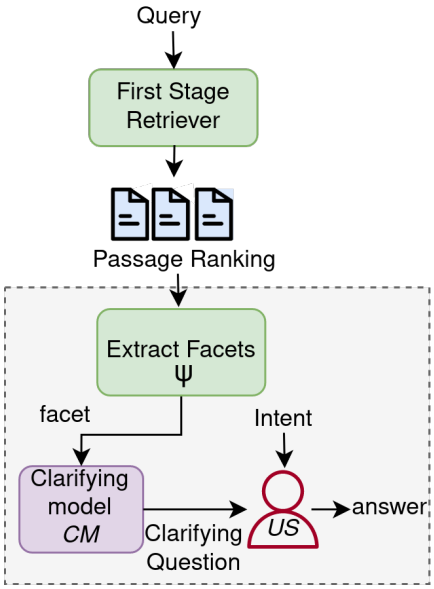


FIGURE 1 – Méthodologie d'évaluation en ligne pour créer des interactions à initiative mixte sur le jeu de test

$$a = \begin{cases} 'yes' \; si \; f \in \mathcal{F}^+ \\ 'no' \; sinon \end{cases} \tag{2}$$

Nous proposons une méthodologie en 3 étapes présentée dans l'Algorithme 1. Étant donné une requête q : 1) des facettes positives et négatives, respectivement $\mathcal{F}^+$ et $\mathcal{F}^-$, sont extraites des ensembles de passages pertinents et non pertinents, respectivement $\mathcal{P}_q^+$ et $\mathcal{P}_q^-$; 2) une interaction X_i est émise pour une facette f, générant la question de clarification associée cq (avec $\mathcal{CM}$) et associant la réponse a à la pertinence de la facette (équation 2) ; 3) l'ensemble d'interactions X est incrémenté avec cette nouvelle interaction X_i, ce qui permet de construire un jeu de données de RI à initiative mixte en associant l'ensemble d'interactions X construit sur toutes les requêtes du jeu de données de RI ad hoc initial $\mathcal{D}$.

3.4.2 Construction des données de test pour l'évaluation en ligne sans jugement de pertinence

Notre méthodologie en ligne vise à générer des interactions sans s'appuyer sur la pertinence des documents. Au lieu de cela, nous tirons parti de la rétroaction de pseudo-pertinence en utilisant les SERPs d'un modèle de recherche d'information comme un proxy pour extraire les facettes de la requête. Chaque facette conditionne la génération de la question de clarification et de la réponse. Plus particulièrement, le pipeline proposé pour générer des interactions en ligne pour une requête q est présenté dans la figure 1. Il est construit sur les étapes suivantes : 1) classement des documents à l'aide d'un modèle de recherche d'information (dans notre cas BM25), 2) extraction de l'ensemble des facettes sur la base des documents pseudo-pertinent, et 3) génération de l'interaction.

Selon les besoins de l'évaluation, différents choix peuvent être faits concernant l'extraction des

facettes. On peut extraire une seule facette du document le mieux classé afin d'effectuer une seule étape de recherche pour une requête (la stratégie utilisée dans nos expériences). D'autres tâches ou objectifs d'évaluation nécessiteraient la génération de multiples facettes et, par conséquent, de multiples clarifications. Cela peut être fait en identifiant les documents les plus hauts/les plus bas obtenus avec le classement de la première étape comme des documents pseudo-pertinents ; chaque document conditionnant l'extraction de facettes comme décrit dans la section 3.2.

4 Évaluation de la méthodologie de génération

Dans cette section, nous évaluons notre méthodologie, et en particulier, la qualité des interactions simulées. Veuillez noter que nous nous concentrons sur l'augmentation du jeu de données MsMarco, mais que notre méthodologie peut être généralisée à tout jeu de données de RI ad hoc.

4.1 Protocole d'Evaluation

4.1.1 Jeux de Données

Nous nous concentrons ici sur le jeu de données MsMarco 2021 passages (Nguyen *et al.*, 2016) qui est un jeu de données de RI à domaine ouvert contenant 8,8 millions de passages et plus de 500 000 paires de pertinence requête-passage avec environ 1,1 passage pertinent par requête en moyenne. MsMarco est couramment utilisé pour entrainer et évaluer les architectures de recherche d'information (Thakur *et al.*, 2021). Nous tirons parti des passages de MsMarco pour extraire des exemples négatifs et donc nos tuples $(q, \mathcal{P}^+, \mathcal{P}^-)$ en s'appuyant sur des modèles de l'état de l'art (Reimers & Gurevych, 2019) [1]. Pour entraîner le modèle de clarification $\mathcal{CM}$, nous utilisons la version filtrée du jeu de données ClariQ proposé dans (Sekulić *et al.*, 2021) qui associe des questions de clarification à des facettes. Toutes les questions de clarification de ce jeu de données sont construites de manière à attendre des réponses " oui " ou " non ". Ce jeu de données fournit 1756 tuples supervisés de (requête-facette-question clarificatrice) pour 187 requêtes.

Pour entraîner la simulation utilisateur $\mathcal{US}$, nous exploitons la moitié de l'ensemble d'entraînement du jeu de données MsMarco (250000 requêtes) pour extraire les facettes positives et négatives comme détaillé dans la section 3.2 et générer des questions de clarification en utilisant le modèle $\mathcal{CM}$. L'étiquette de supervision liée aux réponses est déduite comme proposé dans l'évaluation hors ligne (voir l'équation 2).

Pour l'évaluation hors ligne, étant donné que le jeu de données original comprend des annotations éparses, c'est-à-dire que certains passages sont pertinents mais ne sont pas annotés en tant que tels, il est possible que des documents pertinents soient considérés comme des documents non pertinents. Cette tendance se retrouve toutefois dans l'ensemble d'entraînement MsMarco qui ne comprend qu'un seul document pertinent par requête. Par conséquent, pour assurer la cohérence de l'étiquetage, nous suivons (Qu *et al.*, 2021) et débruitons les exemples négatifs dans l'ensemble du jeu d'entraînement en utilisant un modèle d'encodeur croisé pré-entraîné [2] qui capture les similarités entre les passages. Pour l'évaluation en ligne, nous avons choisi de générer une seule interaction basée sur le document le mieux classé afin de correspondre à notre tâche d'évaluation extrinsèque basée sur la RI. Nous publierons, après acceptation, les jeux de données générés complets ainsi que le modèle de clarification $\mathcal{CM}$ et la simulation utilisateur $\mathcal{US}$ pour permettre la génération d'interactions supplémentaires. Le tableau 1 présente quelques exemples de conversations simulées générées à partir de requêtes MsMarco.

1. https ://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives
2. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

4.1.2 Modèles de référence et métriques

Évaluation des questions de clarification avec des métriques automatiques. Nous nous appuyons sur (Sekulić *et al.*, 2021) et comparons notre modèle de clarification, noté $\mathcal{CM}$, avec 1) une approche basée sur un modèle ($Template$). Le modèle suit une séquence prédéfinie concaténant des facettes : 'Are you looking for + Facet'. 2) $\mathcal{CM}w/oFacet$: la version de notre modèle $\mathcal{CM}$ uniquement conditionnée par la requête. Il s'agit en fait d'un modèle T5 entraîné comme modèle de traduction automatique, qui génère une question de clarification à partir de la requête uniquement. Nous évaluons la capacité de $\mathcal{CM}$ à générer des questions de clarification en utilisant les références fournies dans l'ensemble de test ClariQ. Nous prenons en compte la métrique METEOR (Banerjee & Lavie, 2005) et la similarité moyenne en cosinus entre les phrases incorporées (COSIM). METEOR est couramment utilisé pour évaluer les résultats de la traduction automatique en considérant le rappel et la précision unigramme. Au niveau de la phrase, cette méthode présente une bonne corrélation avec les jugements humains (Banerjee & Lavie, 2005). Pour calculer le score de similarité, nous encodons les questions à l'aide d'un transformer MiniLM-L6-v2 bien entraîné (Reimers & Gurevych, 2019). Nous utilisons le t-test pour évaluer la significativité des différences (*** : valeur p<0,005). Pour évaluer si les questions générées sur MsMarco sont similaires à leur passage relatif, nous calculons également la similarité cosinus moyenne entre les questions de clarification et leurs passages pertinents et non pertinents récupérés. Nous encodons les questions en utilisant MiniLM-L6-v2 (Reimers & Gurevych, 2019).

Évaluations humaines sur des questions de clarification. Pour comparer et mieux évaluer la qualité d'une question de clarification générée sur MsMarco, nous avons effectué une évaluation humaine. À partir de la requête initiale de l'utilisateur et du passage utilisé pour générer la question, nous avons demandé aux annotateurs d'évaluer la qualité de 200 questions de clarification échantillonnées parmi les trois modèles ($Template$, $\mathcal{CM}w/oFacets$ et notre modèle $\mathcal{CM}$). Pour ce faire, les annotateurs sont invités à sélectionner une question de clarification préférée parmi les trois suggestions affichées dans un ordre mélangé pour les critères suivants : **1) L'Utilité** qui évalue si une question peut aider à mieux comprendre ou à affiner la requête en fournissant des informations ou des suggestions supplémentaires. **2) Le Naturel** qui évalue la fluidité et la lisibilité de la question. **3) La Pertinence** qui évalue si une question est spécifique ou liée aux informations contenues dans un passage. Chaque annotateur a évalué 20 cas différents et, pour chaque métrique, a identifié le meilleur modèle de sortie. Nous avons recruté 10 évaluateurs. Chaque instance est évaluée par 2 annotateurs et nous obtenons une métrique Kappa égale à 0,324 montrant un accord équitable entre les évaluateurs. Nous avons également distingué les résultats pour les réponses positives et négatives des utilisateurs en échantillonnant les facettes pertinentes et non pertinentes.

Évaluations humaines sur les réponses. Une hypothèse forte de notre méthode est que les questions de clarification générées avec des facettes extraites de passages pertinents conduisent à des réponses positives alors que l'utilisation de passages non pertinents pour générer des facettes négatives conduit intrinsèquement à des réponses négatives. Pour valider cette hypothèse forte, nous avons montré à des évaluateurs humains différentes instances comprenant une requête q, une question de clarification cq, et le passage pertinent p utilisé pour construire la facette f. Pour chaque instance, nous avons demandé aux évaluateurs humains de répondre par 'oui' ou 'non' aux questions de clarification. Cette évaluation humaine implique 10 annotateurs humains pour un total de 200 questions, avec un équilibre entre les facettes pertinentes et non pertinentes utilisées pour générer la question de clarification. Chaque instance est annotée par deux humains. Nous obtenons une métrique de Kappa égale à 0,472 montrant un accord modéré entre les évaluateurs. Pour valider notre hypothèse, nous considérons les

réponses humaines comme référence et nous les comparons à notre méthode d'étiquetage automatique (à savoir, la simulation utilisateur $\mathcal{US}$) pour calculer la métrique de précision.

4.1.3 Détails d'implémentation

Pour les deux modèles $\mathcal{CM}$ et $\mathcal{US}$, nous avons utilisé le point de contrôle T5 pré-entraîné disponible sur le hub Huggingface (Raffel *et al.*, 2020; Wolf *et al.*, 2019). Pour affiner ces deux modèles, nous avons utilisé le Teacher Forcing (Williams & Zipser, 1989) , nous utilisons AdaFactor (Shazeer & Stern, 2018), et un taux d'apprentissage de 5.10^{-5} avec des tailles de batch de 64. L'incorporation des mots est calculée à l'aide d'un modèle MiniLM-L6-v2 pré-entraîné (Reimers & Gurevych, 2019). Le nombre de mots extraits est fixé à $k = 5$ pour l'ensemble des expériences. Pour l'inférence, nous utilisons l'échantillonnage du noyau (p=0,95) pour les modèles $\mathcal{CM}$ et $\mathcal{US}$.

4.2 Évaluation des interactions générées

4.2.1 Évaluation Automatique

Le tableau 2 rapporte l'efficacité du modèle de clarification sur l'ensemble de test ClariQ. Les résultats montrent que notre modèle $\mathcal{CM}$ surpasse significativement tous les modèles de référence. Les résultats inférieurs obtenus par le modèle de référence $\mathcal{CM}w/oFacet$ soulignent que modèle sans facette est moins efficace que les modèles utilisant des facettes. Les facettes sont utiles pour contraindre le modèle de clarification, et les modèles seq-to-seq basés sur de grands modèles de langage sont plus naturels que les méthodes basées sur des templates. Les facettes sont extraites d'un passage pertinent ou non pertinent et utilisées pour générer des questions de clarification. Le tableau 3 rapporte la similarité en cosinus entre les questions encodées et les passages respectifs. Nous observons que la similarité entre les questions de clarification et leurs passages connexes (en gras) est plus élevée que celle entre les questions de clarification et les requêtes. Cela montre que les questions générées ne sont pas génériques à la requête mais orientées vers les passages fournis.

	METEOR	COSIM
$Template$	0.338^{***}	0.643^{***}
$\mathcal{CM}w/oFacet$	0.326^{***}	0.608^{***}
$\mathcal{CM}$	**0.557**	**0.812**

TABLE 2 – Évaluation des questions de clarification sur le test de ClariQ. *** pour les résultats significatifs avec le modèle $\mathcal{CM}$ (p<0.005)

	q	p+	p-
cq+	0.675	**0.721**	0.450
cq-	0.521	0.450	**0.685**

TABLE 3 – Similarité cosinus entre les questions et leur passage respectif. $cq+$, $cq-$ pour les questions positives et negatives.

4.2.2 Évaluation Humaine

Nous présentons l'évaluation humaine des questions de clarification dans le tableau 4. Le modèle $\mathcal{CM}$ sans facette génère des questions plus naturelles que les autres modèles (préféré pour 46.3% de l'échantillon). Le modèle $\mathcal{CM}$ ajusté avec facette génère plus de questions utiles et pertinentes, ce modèle est considéré comme le plus pertinent par les évaluateurs dans $59,9\%$ de l'échantillon testé. Cela montre que la facette récupérée dans la génération aide à générer des questions plus utiles et plus pertinentes.

Dans l'évaluation humaine des réponses, nous obtenons une précision de 0,685 entre les réponses humaines et l'étiquetage automatique des questions de clarification. Il existe de multiples causes expliquant la différence entre les réponses humaines et l'étiquetage automatique. 1) Facette ne capture pas toujours correctement les informations fournies dans un passage, ce qui conduit à des questions

	Answer	Naturalness	Usefulness	Relevance
Template	positive	0.044	0.086	0.120
	negative	0.073	0.095	0.146
	total	0.119	0.181	0.267
$\mathcal{CM}w/oFacet$	positive	**0.243**	0.195	0.077
	negative	**0.220**	0.140	0.056
	total	**0.463**	0.336	0.133
$\mathcal{CM}$	positive	0.206	**0.213**	**0.297**
	negative	0.211	**0.268**	**0.301**
	total	0.417	**0.481**	**0.599**

TABLE 4 – Résultats de l'évaluation humaine sur Msmarco-passage. Le $\mathcal{CM}$ sans facette produit des questions plus naturelles, mais pas aussi pertinentes que le $\mathcal{CM}$ avec facette.

de clarification de mauvaise qualité. 2) Le modèle $\mathcal{CM}$ ne génère pas toujours une question orientée vers la facette fournie et produit une reformulation de la requête initiale, posant ainsi une question non liée à une facette.

5 Évaluation sur une tâche de RI

Dans cette section, nous proposons d'évaluer indirectement la qualité du jeu de données généré à travers une tâche de RI. En effet, des travaux précédents (Qu *et al.*, 2020; Zhou *et al.*, 2020b; Li *et al.*, 2018a; Fu *et al.*, 2020; Jia *et al.*, 2022) ont déjà utilisé des tâches extrinsèques pour valider un jeu de données. Par conséquent, nous introduisons un modèle de recherche d'information neuronal qui estime les scores de pertinence des passages en fonction de la requête et d'une interaction d'initiative mixte. Notre objectif est double : 1) L'application de ce modèle à notre jeu de données généré permet de savoir si la question de clarification et la réponse associée donnent effectivement un retour utile pour mieux comprendre le besoin d'information sous-jacent. L'évaluation est basée sur l'hypothèse suivante : si un modèle de RI utilisant les interactions générées est plus performant que celui qui ne les utilise pas, les interactions sont considérées comme pertinentes et utiles. 2) Nous fournissons une première base de modèles de référence pour les tâches de RI à initiative mixte.

5.1 Modèle de RI tirant parti des interactions d'initiative mixte

Nous proposons un modèle simple basé sur une architecture d'encodeurs croisés qui s'est avéré efficace pour les tâches de RI, en particulier lors de l'utilisation de modèles de langage de grande taille : (Pradeep *et al.*, 2021). L'encodeur croisé précédent vise à prédire la pertinence d'un passage p étant donné une requête q $P(relevant = 1|q, p)$. Notre modèle estime un score pour les passages en fonction de la requête, d'une question de clarification et d'une réponse de l'utilisateur (q, cq, a) :

$$p(relevant = 1|p, q, cq, a) \tag{3}$$

Suivant (Pradeep *et al.*, 2021), le score de pertinence est calculé grâce la log-probabilité prédite des tokens vrai/faux :

$$s_p = \log p(true|q, p, cq, a) \tag{4}$$

Suivant (Pradeep *et al.*, 2021), nous utilisons le modèle MonoT5 et intégrons des interactions d'initiative mixte en plus de la requête initiale pour mieux d'estimer les scores des documents. La séquence d'entrée est une concaténation de la requête, du document, de la question et de la réponse, séparés par des tokens spéciaux :

$$\text{Query : q Document : d Question : cq Answer : a} \tag{5}$$

	MRR@10	NDCG@1	NDCG@3	NDCG@10
BM25	0.1840***	0.105***	0.1690***	0.228***
BM25 + RM3	0.1566***	0.0807***	0.1386***	0.2021***
BM25 + MonoT5	0.3522***	0.2398***	0.3457***	0.4034***
BM25 + CLART5	**0.3863**	**0.2788**	**0.3817**	**0.4327**

TABLE 5 – Performance de RI sur MiMarco test. *** : two-sided t-test w.r.t. BM25+CLART5. with p-value<0.005

5.2 Détails d'entraînement

Nous avons utilisé MonoT5 pré-entraîné disponible sur le hub Huggingface (Raffel *et al.*, 2020; Wolf *et al.*, 2019). Nous affinons ce modèle sur notre ensemble d'entraînement, en utilisant teacher forcing et entropie croisée. Nous considérons une longueur de séquence maximale de 512 et une taille de batch de 128 séquences. Afin d'apprendre correctement à faire la distinction entre les passages pertinents et non pertinents d'une question, nous intégrons les interactions négatives dans le batch.

Pour l'optimisation, nous utilisons AdaFactor (Shazeer & Stern, 2018), et un taux d'apprentissage de 10^{-4}. Le réglage fin du modèle prend environ 4 heures sur 4 RTX 3080 (24 Go).

Au moment du test, nous effectuons une recherche de documents sur la requête initiale en utilisant l'implémentation pyserini (Lin *et al.*, 2021) de BM25. Nous appliquons ensuite notre modèle comme un modèle d'ordonnancement avec des informations supplémentaires. Nous fixons le nombre de documents récupérés à 100.

5.3 Métriques et modèles de référence

Nous utilisons des mesures classiques pour évaluer la qualité de l'ordonnancement des documents, à savoir le gain cumulé actualisé normalisé (NDCG) aux rangs 1, 3 et 10, et le rang réciproque moyen (MRR) au rang 10. Pour évaluer le potentiel de notre jeu de données, nous comparons les performances de notre modèle, noté **BM25+CLART5**, aux approches suivantes :

— **BM25.** BM25 est un modèle d'ordonnancement connu qui s'appuie sur la fréquence des mots contenu dans les documents, couramment utilisé comme référence.

— **BM25 + RM3.** RM3 est une méthode de pseudo-pertinence pour l'expansion de requête. La requête est développée à l'aide de termes d'expansion extraits des 10 premiers documents récupérés. RM3 est une base de référence compétitive et est souvent utilisée pour évaluer les modèles de RI. (Thakur *et al.*, 2021; Adolphs *et al.*, 2022).

— **BM25 + MonoT5.** MonoT5 est un modèle d'ordonnancement pré-entraîné sur l'ensemble d'entraînement original de MsMarco, c'est-à-dire uniquement les requêtes et les jugements de pertinence. Ce modèle atteint des performances de pointe sur le tableau de classement beir (Thakur *et al.*, 2021) et constitue une référence naturelle puisque BM25+CLART5 utilise le même modèle pré-entraîné de deuxième étape avant de l'affiner sur des interactions.

5.4 Efficacité de l'ordonnancement neuronal orienté initiative mixte

Nous présentons les résultats de notre modèle d'ordonnancement neuronal à initiative mixte obtenus sur le pipeline d'évaluation en ligne présenté dans la section 3.4 appliqué sur l'ensemble de test MsMarco (Tableau 5).

Le tableau 5 met en évidence le fait que les informations supplémentaires permettent à BM25+CLART5 d'améliorer de manière significative toutes les métriques sur le jeu de données MsMarco augmenté. Par exemple, BM25+CLART5 augmente le score MRR@10 de 0,034 point par

rapport à BM25+MonoT5. Une analyse plus poussée des résultats sur MsMarco montre que pour 33.0% des requêtes, le passage pertinent n'est pas récupéré dans le top-100 par BM25, conduisant le MRR@100 à 0.0. Pour 25,6 des requêtes, Monot5 et ClarT5 obtiennent la même valeur MRR@10. Pour $30,3\%$des requêtes, BM25+CLART5 obtient un meilleur MRR@10 tandis que $11,1\%$ obtiennent un MRR@10 inférieur. Dans l'ensemble, ces résultats montrent que le feedback fourni par la simulation de l'utilisateur à la question de clarification est pertinent et utile. Il permet d'augmenter le classement des passages pertinents. Ce résultat confirme indirectement que les interactions simulées encodent effectivement des informations pertinentes pour les intentions de recherche sous-jacentes, ce qui correspond à ce que les utilisateurs réels fourniraient dans les conversations. Par conséquent, les simulations proposées sont raisonnables.

6 Expériences complémentaires

6.1 Extension aux interactions multi-tours

Dans la section précédente, nous avons simulé une interaction avec une seule requête $X = (q, cq, a)$ pour l'inférence en ligne. Cependant, de multiples facettes différentes peuvent être extraites des passages récupérés. Cela signifie que des séquences d'interactions $X_0, ..., X_t$ peuvent être inférées en sélectionnant séquentiellement différentes facettes. Bien qu'un nouvel ensemble de passages puisse être récupéré en utilisant la dernière interaction, nous ne considérons ici que les facettes des passages récupérés avec la requête initiale. Chaque t^{ieme} tour exploite le t^{ieme} document dans la liste de documents pour construire une facette et générer une question de clarification. Les interactions multi-tours sont donc générées dans un ordre non arbitraire.

Impact sur la conception du modèle de RI neuronal. Nous proposons d'étendre le modèle au re-classement multi-tour en utilisant plusieurs tours de clarification autour de la même requête. Nous évaluons les passages en utilisant des interactions multiples autour de la même intention de recherche. À chaque pas de temps t, un nouveau score s_d^t est calculé pour les passages du même classement utilisant une seule interaction. Ce score est calculé en utilisant l'équation 6 qui prédit les scores de pertinence cumulatifs à toutes les interactions, c'est-à-dire la somme des scores de pertinence jusqu'au temps T. Ce score est utilisé comme le score d'un document suivant une séquence d'interactions $X_t = \{q, cq^1, a^1, ..., cq^t, a^t\}$. cq^t et a^t sont la question de clarification et la réponse générées à l'instant t.

$$s_d^T = \sum_{t=0}^{T} \log p(relevant = 1 | q, p, cq^t, a^t) \tag{6}$$

où s_d^T est le score du document p au temps T. Comme le classement est mis à jour entre les tours, nous sélectionnons les facettes du passage le mieux classé à chaque pas de temps. Nous évaluons la performance d'ordonnancement à différentes longueurs d'interactions, de $T = 1$ à $T = 5$. Nous présentons également l'entropie du classement (Shannon, 1948) comme une mesure de la confiance du système en mesurant comment les scores sont distribués dans le classement. Cette entropie est maximisée lorsque la distribution des scores est uniforme sur le classement.

Resultats. Le tableau 6 présente les performances en fonction du nombre de tours de clarification pour le jeu de test MsMarco. Nous observons que l'augmentation du nombre de tours améliore les scores MRR, NDCG et tout en minimisant l'entropie, ce qui suggère que les informations supplémentaires aident à affiner davantage les besoins d'informations de l'utilisateur.

	MRR@10	NDCG@1	NDCG@3	NDCG@10	Entropy
BM25 + CLART5 T=1	0.3863	0.2788	0.3817	0.4327	2.951
BM25 + CLART5 T=2	0.44467	0.35186	0.43734	0.48038	2.303
BM25 + CLART5 T=3	0.48176	0.39828	0.47483	0.51089	2.163
BM25 + CLART5 T=4	0.50861	0.43266	0.50321	0.53183	2.06
BM25 + CLART5 T=5	**0.52949**	**0.45946**	**0.52311**	**0.54814**	**1.99**

TABLE 6 – Résultats d'efficacité sur les tours d'interaction T émis sur les ensembles de test MsMarco.

Multi-turn Conversations	**mrr@100**	**Entropy**
User : where was the bauhaus built	0.04167	3.854
System : do you want to know where the bauhaus was founded ? **User** : no	0.25	3.529
System : do you want to know where the gropius sommerfeld house was built ? **User** : yes	0.5	0.693
System : do you want to know where the bauhaus art school was built ? **User** : no	1.0	0.000218
User : what are tor	0.0	3.776
System : do you want to know if tor is free ? **User** : no	0.0	3.612
System : do you want to know more about tors transoral robotic surgery ? **User** : no	0.0	3.586
System : do you want to know how to connect to a computer network **User** : no	0.0	3.567
System : are you looking for a web browser for tor onion ? **User** : no	0.0	3.541
System : do you want to know about tor anonymity ? **User** : no	0.0	3.329

TABLE 7 – Exemples de cas réussis (en haut) et de cas d'échec (en bas). Dans le cas du haut, les questions de clarification aident à affiner l'intention de l'utilisateur et à mieux identifier les passages pertinents. Cet exemple montre une ambiguïté typique dans le document récupéré : le Bauhaus peut faire référence au groupe de rock, à l'école d'art, au prix... Pour le cas d'échec, aucun des passages pertinents n'est récupéré, posant des questions de clarification et reclassant les passages n'améliore pas les scores.

Nous décrivons des exemples qualitatifs de séquences de clarification réussies et non réussiesdans la table 7. Nous pouvons voir dans le premier exemple qu'une interaction supplémentaire permet de mieux affiner les scores du passage pertinent conduisant à un meilleur MRR@100, tandis que l'entropie diminue. Dans la dernière interaction, l'entropie est très faible, ce qui signifie que la distribution des scores est dense sur quelques passages. D'autre part, le deuxième exemple est un cas d'échec où les passages pertinents ne sont même pas récupérés. Dans certain cas d'échec que nous observons, où l'interaction tourne détériorer le classement, ce qui montre que les interactions générées ne sont pas toujours parfaites.

7 Conclusion et discussion

Il existe un besoin critique d'ensembles de données adéquats avec des interactions à initiative mixte pour la RI conversationnelle, mais la création d'un tel jeu de données est très coûteuse. La collecte à grande échelle de donnée de recherche conversationnelle interactive mixte dans un domaine ouvert avec un jugement de pertinence de document annoté reste très coûteuse. Dans cet article, nous avons proposé une méthode pour augmenter les ensembles de données IR ad hoc en simulant une forme simple d'interactions de clarification entre un utilisateur et un système. Cette méthode génère automatiquement les questions et les réponses à partir d'un grand jeu de données RI, permettant

d'expérimenter des approches RI conversationnelles à grande échelle. L'approche proposée est générique et peut être appliquée à tout jeu de données RI ad hoc existant. Dans les expériences, nous avons augmenté le jeu de données MsMarco et évalué la qualité des interactions avec les tâches intrinsèques et extrinsèques, en nous appuyant sur des métriques automatiques et des évaluations humaines. Les résultats montrent que, malgré la simplicité de nos approches, les interactions générées sont pertinentes pour les intentions de recherche et utiles pour un meilleur classement des documents. De plus, nous étendons notre méthodologie à un cadre de clarification multi-tours et fournissons des expériences préliminaires mettant en évidence le potentiel de notre méthodologie. Il s'agit d'une première approche pour l'augmentation d'ensembles de données à grande échelle pour la RI conversationnelle. Il démontre la faisabilité de la construction automatique de jeux de données. En tant que première investigation, cette étude présente plusieurs limites qui pourront être améliorées dans le futur.

— Dans un premier temps, notre investigation se limite à clarifier des questions basées sur une seule facette, souvent assimilées à des questions du type : "Fais-tu référence à 'facette' ?". Cependant, de véritables questions de clarification peuvent également poser des questions sur plusieurs sujets/facettes en un seul tour (ex : Êtes-vous intéressé à connaître *sujet1*, *sujet2* ou *sujet3*) ou également être formulées comme ouvertes questions terminées (par exemple, "Qu'aimeriez-vous savoir sur *sujet* ?"). Ces questions plus complexes sont plus difficiles à générer et à répondre dans les simulations, mais peut potentiellement apporter plus d'informations et être plus naturel dans la conversation.
— Deuxièmement, l'extraction des facettes reposait sur quelques mots-clés et cela peut être amélioré. Nous observons que lorsque les passages sont longs et traitent de plusieurs sujets, la question générée peut ne pas représenter le sujet abordé dans le passage. L'extraction des facettes doit être améliorée.
— Troisièmement, la simulation de l'utilisateur a été limitée aux réponses 'oui'/'non'. Dans une recherche conversationnelle plus sophistiquée, l'utilisateur pourrait fournir des informations plus et diverses dans la réponse. Simuler des réponses d'utilisateurs plus complexes est un défi pour l'avenir.
— Enfin, nous avons également généré des interactions multi-tours mais n'avons pas considéré la dépendance entre les tours. Dans une recherche conversationnelle réelle, les tours ultérieurs peuvent dépendre des précédents. Des simulations plus raisonnables d'interactions multitours devraient tenir compte de la dépendance. Ce cadre doit être profondément réfléchi. En effet, les conversations générées ne sont qu'une concaténation de plusieurs tours de clarification indépendants autour d'une même requête utilisateur. Il est crucial de définir une stratégie pour sélectionner la bonne séquence de questions de clarification afin d'optimiser la réussite de la session de recherche.

Malgré les limites, la démonstration de faisabilité faite dans cet article pour créer des jeux de données de RI conversationnelles à grande échelle ouvre la porte à d'autres enquêtes à grande échelle sur le sujet. Cela dit, nous espérons cependant que notre méthodologie aiderait la communauté à définir des cadres d'évaluation pour la RI conversationnelle en tirant parti des jeux de données RI ad hoc existants.

Remerciements

Nous tenons à remercier le projet ANR JCJC SESAMS (Projet-ANR-18-CE23-0001) pour son soutien à Pierre Erbacher et Laure Soulier de Sorbonne Université dans le cadre de ce travail.

Références

ADOLPHS L., HUEBSCHER M. C., BUCK C., GIRGIN S., BACHEM O., CIARAMITA M. & HOFMANN T. (2022). Decoding a neural retriever's latent space for query suggestion. DOI : 10.48550/ARXIV.2210.12084.

ALIANNEJADI M., KISELEVA J., CHUKLIN A., DALTON J. & BURTSEV M. (2021). Building and evaluating open-domain dialogue corpora with clarifying questions. In *EMNLP*.

ALIANNEJADI M., ZAMANI H., CRESTANI F. & CROFT W. B. (2019). Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, p. 475–484, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3331184.3331265.

ANAND A., CAVEDON L., HAGEN M., JOHO H., SANDERSON M. & STEIN B. (2020). Conversational search - a report from dagstuhl seminar 19461. *CoRR*, **abs/2005.08658**.

ASRI L. E., HE J. & SULEMAN K. (2016). A sequence-to-sequence model for user simulation in spoken dialogue systems. DOI : 10.48550/ARXIV.1607.00070.

BANERJEE S. & LAVIE A. (2005). METEOR : An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, p. 65–72, Ann Arbor, Michigan : Association for Computational Linguistics.

BELKIN N. J. & CROFT W. B. (1992). Information filtering and information retrieval : Two sides of the same coin ? *Commun. ACM*, **35**(12), 29–38. DOI : 10.1145/138859.138861.

BENAMARA F., HATOUT N., MULLER P. & OZDOWSKA S., Éds. (2007). *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse. ATALA, IRIT.

BENDER E. M., GEBRU T., MCMILLAN-MAJOR A. & SHMITCHELL S. (2021). On the dangers of stochastic parrots : Can language models be too big ? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, p. 610–623, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3442188.3445922.

BI K., AI Q. & CROFT W. B. (2021). Asking clarifying questions based on negative feedback in conversational search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, p. 157–166, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3471158.3472232.

CHEN D., FISCH A., WESTON J. & BORDES A. (2017). Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1870–1879, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1171.

CHOI E., HE H., IYYER M., YATSKAR M., YIH W.-T., CHOI Y., LIANG P. & ZETTLEMOYER L. (2018). QuAC : Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2174–2184, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1241.

CHU-CARROLL J. & BROWN M. K. (1997). Tracking initiative in collaborative dialogue interactions. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, p. 262–270, Madrid, Spain : Association for Computational Linguistics. DOI : 10.3115/976909.979651.

CLARKE C. L. A., CRASWELL N. & SOBOROFF I. (2009). Overview of the TREC 2009 web track. In E. M. VOORHEES & L. P. BUCKLAND, Éds., *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009*, volume 500-278 de *NIST Special Publication* : National Institute of Standards and Technology (NIST).

DALTON J., FISCHER S., OWOICHO P., RADLINSKI F., ROSSETTO F., TRIPPAS J. R. & ZAMANI H. (2022). Conversational information seeking : Theory and application. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, p. 3455–3458, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3477495.3532678.

DALTON J., XIONG C. & CALLAN J. (2020a). Trec cast 2019 : The conversational assistance track overview. DOI : 10.48550/ARXIV.2003.13624.

DALTON J., XIONG C., KUMAR V. & CALLAN J. (2020b). *CAsT-19 : A Dataset for Conversational Information Seeking*, In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 1985–1988. Association for Computing Machinery : New York, NY, USA.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

DOU Z., JIANG Z., HU S., WEN J.-R. & SONG R. (2016). Automatically mining facets for queries from their search results. *IEEE Trans. on Knowl. and Data Eng.*, **28**(2), 385–397. DOI : 10.1109/TKDE.2015.2475735.

ELGOHARY A., PESKOV D. & BOYD-GRABER J. (2019). Can you unpack that ? learning to rewrite questions-in-context. 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. 5918–5924, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1605.

ERBACHER P., DENOYER L. & SOULIER L. (2022). : sigir. DOI : 10.48550/ARXIV.2205.15918.

FAN A., JERNITE Y., PEREZ E., GRANGIER D., WESTON J. & AULI M. (2019). ELI5 : Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3558–3567, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1346.

FU Z., XIAN Y., ZHU Y., ZHANG Y. & DE MELO G. (2020). Cookie : A dataset for conversational recommendation over knowledge graphs in e-commerce. DOI : 10.48550/ARXIV.2008.09237.

GAO C., LI S., LEI W., CHEN J., LI B., JIANG P., HE X., MAO J. & CHUA T.-S. (2022). Kuairec : A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information amp ; Knowledge Management*, CIKM '22, p. 540–550, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3511808.3557220.

GLAESE A., MCALEESE N., TRĘBACZ M., ASLANIDES J., FIROIU V., EWALDS T., RAUH M., WEIDINGER L., CHADWICK M., THACKER P., CAMPBELL-GILLINGHAM L., UESATO J., HUANG P.-S., COMANESCU R., YANG F., SEE A., DATHATHRI S., GREIG R., CHEN C., FRITZ D., ELIAS J. S., GREEN R., MOKRÁ S., FERNANDO N., WU B., FOLEY R., YOUNG S., GABRIEL I., ISAAC W., MELLOR J., HASSABIS D., KAVUKCUOGLU K., HENDRICKS L. A. & IRVING G. (2022). Improving alignment of dialogue agents via targeted human judgements. DOI : 10.48550/ARXIV.2209.14375.

JIA M., LIU R., WANG P., SONG Y., XI Z., LI H., SHEN X., CHEN M., PANG J. & HE X. (2022). E-ConvRec : A large-scale conversational recommendation dataset for E-commerce customer service. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 5787–5796, Marseille, France : European Language Resources Association.

KANG D., BALAKRISHNAN A., SHAH P., CROOK P., BOUREAU Y.-L. & WESTON J. (2019a). Recommendation as a communication game : Self-supervised bot-play for goal-oriented dialogue. DOI : 10.48550/ARXIV.1909.03922.

KANG D., BALAKRISHNAN A., SHAH P., CROOK P., BOUREAU Y.-L. & WESTON J. (2019b). Recommendation as a communication game : Self-supervised bot-play for goal-oriented dialogue. 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. 1951–1961, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1203.

KEYVAN K. & HUANG J. X. (2022). How to approach ambiguous queries in conversational search : A survey of techniques, approaches, tools, and challenges. *ACM Comput. Surv.*, **55**(6). DOI : 10.1145/3534965.

KONG W. & ALLAN J. (2013). Extracting query facets from search results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, p. 93–102, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/2484028.2484097.

KREYSSIG F., CASANUEVA I., BUDZIANOWSKI P. & GAŠIĆ M. (2018). Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, p. 60–69, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5007.

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.

LI R., KAHOU S., SCHULZ H., MICHALSKI V., CHARLIN L. & PAL C. (2018a). Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, p. 9748–9758, Red Hook, NY, USA : Curran Associates Inc.

LI R., KAHOU S. E., SCHULZ H., MICHALSKI V., CHARLIN L. & PAL C. (2018b). Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31 (NIPS 2018)*.

LI X., CHEN Y.-N., LI L., GAO J. & CELIKYILMAZ A. (2017). End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 733–743, Taipei, Taiwan : Asian Federation of Natural Language Processing.

LIN J., MA X., LIN S.-C., YANG J.-H., PRADEEP R. & NOGUEIRA R. (2021). Pyserini : A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, p. 2356–2362.

LIU Z., WANG H., NIU Z.-Y., WU H. & CHE W. (2021). DuRecDial 2.0 : A bilingual parallel corpus for conversational recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 4335–4347, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.356.

MOON S., SHAH P., KUMAR A. & SUBBA R. (2019). OpenDialKG : Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 845–854, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1081.

NAKANO R., HILTON J., BALAJI S., WU J., OUYANG L., KIM C., HESSE C., JAIN S., KOSARAJU V., SAUNDERS W., JIANG X., COBBE K., ELOUNDOU T., KRUEGER G., BUTTON K., KNIGHT M., CHESS B. & SCHULMAN J. (2021). Webgpt : Browser-assisted question-answering with human feedback. DOI : 10.48550/ARXIV.2112.09332.

NGUYEN T., ROSENBERG M., SONG X., GAO J., TIWARY S., MAJUMDER R. & DENG L. (2016). MS MARCO : A human generated machine reading comprehension dataset. *CoRR*, **abs/1611.09268**.

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.

OVER P. (2001). The trec interactive track : an annotated bibliography. *Information Processing Management*, **37**(3), 369–381. Interactivity at the Text Retrieval Conference (TREC), DOI : https ://doi.org/10.1016/S0306-4573(00)00053-4.

PENG B., LI X., GAO J., LIU J. & WONG K.-F. (2018). Deep Dyna-Q : Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2182–2192, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1203.

PRADEEP R., NOGUEIRA R. & LIN J. (2021). The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *CoRR*, **abs/2101.05667**.

QU C., YANG L., CHEN C., QIU M., CROFT W. B. & IYYER M. (2020). Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, p. 539–548, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3397271.3401110.

QU Y., DING Y., LIU J., LIU K., REN R., ZHAO W. X., DONG D., WU H. & WANG H. (2021). RocketQA : An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 5835–5847, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.466.

RADLINSKI F. & CRASWELL N. (2017). A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, p. 117–126, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3020165.3020183.

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.

RAJPURKAR P., JIA R. & LIANG P. (2018). Know what you don't know : Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 784–789, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-2124.

REDDY S., CHEN D. & MANNING C. D. (2019). CoQA : A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, **7**, 249–266. DOI : 10.1162/tacl_a_00266.

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* : Association for Computational Linguistics.

SALLE A., MALMASI S., ROKHLENKO O. & AGICHTEIN E. (2021). Studying the effectiveness of conversational search refinement through user simulation. In D. HIEMSTRA, M.-F. MOENS, J. MOTHE, R. PEREGO, M. POTTHAST & F. SEBASTIANI, Éds., *Advances in Information Retrieval*, p. 587–602, Cham : Springer International Publishing.

SCHATZMANN J., THOMSON B., WEILHAMMER K., YE H. & YOUNG S. (2007). Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007 : The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, p. 149–152, Rochester, New York : Association for Computational Linguistics.

SEKULIĆ I., ALIANNEJADI M. & CRESTANI F. (2021). Towards facet-driven generation of clarifying questions for conversational search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, p. 167–175, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3471158.3472257.

SEKULIĆ I., ALIANNEJADI M. & CRESTANI F. (2022). Exploiting document-based features for clarification in conversational search. In M. HAGEN, S. VERBERNE, C. MACDONALD, C. SEIFERT, K. BALOG, K. NØRVÅG & V. SETTY, Éds., *Advances in Information Retrieval*, p. 413–427, Cham : Springer International Publishing.

SERETAN V. & WEHRLI E. (2007). Collocation translation based on sentence alignment and parsing. In (Benamara *et al.*, 2007), p. 401–410.

SHAH C. & BENDER E. M. (2022). Situating search. In *ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '22, p. 221–232, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3498366.3505816.

SHANNON C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, **27**, 379–423.

SHARMA P. & LI Y. (2019). Self-supervised contextual keyword and keyphrase retrieval with self-labelling. DOI : 10.20944/preprints201908.0073.v1.

SHAZEER N. & STERN M. (2018). Adafactor : Adaptive learning rates with sublinear memory cost. In J. DY & A. KRAUSE, Éds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, p. 4596–4604 : PMLR.

SHUSTER K., KOMEILI M., ADOLPHS L., ROLLER S., SZLAM A. & WESTON J. (2022). Language models that seek for knowledge : Modular search amp ; generation for dialogue and prompt completion. DOI : 10.48550/ARXIV.2203.13224.

THAKUR N., REIMERS N., RÜCKLÉ A., SRIVASTAVA A. & GUREVYCH I. (2021). BEIR : A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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.

TRIPPAS J. R., SPINA D., THOMAS P., SANDERSON M., JOHO H. & CAVEDON L. (2020). Towards a model for spoken conversational search. *Information Processing Management*, **57**(2), 102162. DOI : https ://doi.org/10.1016/j.ipm.2019.102162.

WILLIAMS R. J. & ZIPSER D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, **1**(2), 270–280. DOI : 10.1162/neco.1989.1.2.270.

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., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2019). Huggingface's transformers : State-of-the-art natural language processing. DOI : 10.48550/ARXIV.1910.03771.

WU F., QIAO Y., CHEN J.-H., WU C., QI T., LIAN J., LIU D., XIE X., GAO J., WU W. & ZHOU M. (2020). MIND : A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 3597–3606, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.331.

ZAMANI H., DUMAIS S., CRASWELL N., BENNETT P. & LUECK G. (2020a). *Generating Clarifying Questions for Information Retrieval*, In *Proceedings of The Web Conference 2020*, p. 418–428. Association for Computing Machinery : New York, NY, USA.

ZAMANI H., MITRA B., CHEN E., LUECK G., DIAZ F., BENNETT P. N., CRASWELL N. & DUMAIS S. T. (2020b). Analyzing and learning from user interactions for search clarification. *CoRR*, **abs/2006.00166**.

ZAMANI H., TRIPPAS J. R., DALTON J. & RADLINSKI F. (2022). Conversational information seeking. *CoRR*, **abs/2201.08808**.

ZHOU K., ZHOU Y., ZHAO W. X., WANG X. & WEN J.-R. (2020a). Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain, December 8-11, 2020*.

ZHOU K., ZHOU Y., ZHAO W. X., WANG X. & WEN J.-R. (2020b). Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4128–4139, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.365.

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

Apprentissage de sous-espaces de préfixes

Louis Falissard[1, 2] Vincent Guigue[3] Laure Soulier[1, 4]
(1) Sorbonne Université, CNRS, ISIR, 75005 Paris, France
(2) Bibliothèque nationale de France, 75013 Paris, France
(3) AgroParisTech, UMR MIA-PS, 91120 Palaiseau, France
(4) Université Paris Saclay, CNRS, LISN, 91400 Orsay, France
louis.falissard@gmail.com, vincent.guigue@isir.upmc.fr,
laure.soulier@isir.upmc.fr

RÉSUMÉ

Cet article propose une nouvelle façon d'ajuster des modèles de langue en "Few-shot learning" se basant sur une méthode d'optimisation récemment introduite en vision informatique, l'apprentissage de sous-espaces de modèles. Cette méthode, permettant de trouver non pas un point minimum local de la fonction coût dans l'espace des paramètres du modèle, mais tout un simplexe associé à des valeurs basses, présente typiquement des capacités de généralisation supérieures aux solutions obtenues par ajustement traditionnel. L'adaptation de cette méthode aux gros modèles de langue n'est pas triviale mais son application aux méthodes d'ajustement dites "Parameter Efficient" est quant à elle relativement naturelle. On propose de plus une façon innovante d'utiliser le simplexe de solution étudié afin de revisiter la notion de guidage de l'ajustement d'un modèle par l'inférence d'une métrique de validation, problématique d'actualité en "few-shot learning". On montre finalement que ces différentes contributions centrées autour de l'ajustement de sous-espaces de modèles est empiriquement associée à un gain considérable en performances de généralisation sur les tâches de compréhension du langage du benchmark GLUE, dans un contexte de "few-shot learning".

ABSTRACT

Learning prefix subspaces

This paper proposes a new way of fitting language models in few-shot learning based on a recently introduced optimization method in computer vision, model subspace learning. This method, allowing to find not a local minimum point of the cost function in the parameter space of the model, but a whole simplex associated with low values, typically presents higher generalization capabilities than solutions obtained by traditional fitting. The adaptation of this method to large language models is not trivial, but we observe that its application to so-called "Parameter Efficient" fitting methods is relatively natural. We also propose an innovative way to use the studied solution simplex in order to revisit the notion of guiding the adjustment of a model by inferring a validation metric, a current problem in "few-shot learning". We finally show that these different contributions centered around the adjustment of model subspaces is empirically associated with a considerable gain in generalization performance on the GLUE benchmark language understanding tasks, in a "few-shot learning" context.

MOTS-CLÉS : Modèles de langues, apprentissages sur petits échantillons, apprentissage de sous-espaces, classification de texte.

KEYWORDS: Large language models, few-shot learning, subspace learning, text classification.

Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA), pages 59–73.

1 Introduction

L'avènement au cours des dernières années des gros modèles de langues (Devlin *et al.*, 2019; Radford *et al.*, 2019; Raffel *et al.*, 2019) a été la source d'une évolution considérable des applications de méthodes d'apprentissage profond en traitement automatique des langues. Ces modèles, pré-entraînés de manière non-supervisée sur des corpus de données textuelles massifs, permettent notamment l'ajustement de puissant modèles neuronaux à partir de quelques milliers, voire centaines d'observations, avec des performances de généralisation qui demandaient encore il y a quelques années plusieurs millions d'observations. Plus récemment encore, l'augmentation en dimensionnalité de ces modèles, couplée à des corpus de pré-entraînement encore plus massifs, permet de les utiliser comme base pour l'implémentation de puissant algorithme de classification de texte, ceci avec une poignée d'observations, notamment par leur utilisation en conjonction à des prompts d'instruction discrets (Radford *et al.*, 2019).

L'extension de ces méthodes discrètes à l'apprentissage de prompts différentiables, qui vient s'inscrire au moins conceptuellement dans le cadre des méthodes dites de "Parameter Efficient Fine-Tuning" (PEFT) (Houlsby *et al.*, 2019; Bapna & Firat, 2019), pose cependant quelques problèmes dans un cadre de few-shot learning, dont notamment celui du guidage par la métrique de validation de l'ajustement du modèle pendant l'optimisation par descente. En effet, il est coutume, dans le processus d'ajustement d'un modèle neuronal, d'exclure au préalable environ un tiers des observations du jeux de données d'entraînement afin de créer un jeu dit de validation (ou de développement), consacré à l'inférence d'une estimation non biaisée des performances du modèle. Cette métrique est utilisée autant pendant ajustement du modèle (pour estimer la convergence de l'algorithme de descente, ou pour informer une heuristique d'arrêt prématuré) qu'en aval pour guider la recherche d'hyperparamètres typiquement utilisée dans le cadre d'ajustement de gros modèles de langues. Le bien fondé de cette approche repose en revanche sur la garantie que la distribution du jeu de validation est un minimum représentative de celle du phénomène réel observé. Cette garantie peut très vite perdre de son sens dans un cadre de "few-shot learning", où une dizaine d'observations tout au plus est disponibles à l'estimation de la métrique de validation. Cette notion est de nos jours devenue suffisamment problématique pour qu'une partie de la littérature académique en apprentissage continu sur petits jeux de données présente des résultats d'expériences utilisant des jeux de validations autant irréaliste qu'artificiels, comportant jusqu'à plusieurs ordres de grandeurs plus d'observations que le jeu d'entraînement utilisé pour l'ajustement même du modèle (Wortsman *et al.*, 2021).

Récemment introduit en vision informatique, le concept d'apprentissage de sous-espace de paramètres de modèles (qu'on appellera par soucis de concision "méthode des sous-espaces" (Wortsman *et al.*, 2021) est une technique d'optimisation permettant de trouver non pas minimum local de la fonction coût dans l'espace des paramètre du modèle, mais tout un simplexe associé à de faibles valeurs de cet objectif, comme illustré en figure 1. Les modèles ajustés par le biais de cette méthode présentent notamment des capacités de généralisation supérieures aux solutions obtenues par ajustement traditionnel. Ce phénomène, expliqués empiriquement par les propriétés enviables des minimums locaux qu'elles permettent d'identifier, revêt un intérêt tout particulier lorsqu'on l'observe à travers le prisme d'une problématique de *few-shot learning*, où la capacité du modèle à généraliser une classe de concept à partir d'un nombre réduit d'exemple est clé. Son application directe à l'ajustement de gros modèles de langue, en revanche, n'est pas trivial. En effet, cette méthode de sous-espace propose

d'obtenir ce simplexe de solutions (dans l'espace des paramètres du modèle étudié) par un unique processus de descente de la manière suivante :

— Un simplexe de modèles est initialisé aléatoirement (par initialisation de chacun de ses sommets via une méthode classique d'initialisation non déterministe de modèles neuronaux).
— Un modèle est construit par échantillonnage uniforme sur le simplexe à chaque itération de descente, et utilisé pour l'inférence, le calcul de la fonction objectif et du gradient, de manière à ajuster les sommets du simplexe.

Une fois l'ajustement par descente de gradient terminé, le simplexe peut être utilisé soit dans le cadre de méthodes d'ensemble, soit en choisissant un modèle unique dans le simplexe (généralement son centroïde).

On comprend clairement en se basant sur cette observation pourquoi cette méthode n'a (du moins à notre connaissance) jamais été appliquée en traitement automatique des langues (ou du moins à l'ajustement de modèles de langue). En effet, cette méthode se base fondamentalement sur une initialisation *aléatoire* de l'algorithme de descente, ceci afin de construire un simplexe de modèles initial. En revanche, les modèles de langues pré-entraînés sont par essence initialisés de manière *déterministe*. L'intégralité de leurs capacités de transfert reposent d'ailleurs sur les représentations de données textuelles que ces modèles incorporent dans leur vecteur de paramètre durant le pré-entraînement. De plus, la méthode des sous-espace nécessite de garder en mémoire durant l'ajustement non pas un modèle, mais tous les sommets du simplexes de modèles étudiés. Cette contrainte additionnelle en complexité mémoire est probablement tout à fait gérable dans le cadre de l'ajustement d'un réseau à convolution en vision informatique. Les modèles de langue, en revanche, sont connus pour leur taille considérable pouvant très bien avoisiner la centaine de milliard de paramètres, à tel point que l'ajustement traditionnel de l'un d'entre eux constitue déjà un challenge technique considérable pour la plupart des infrastructures de calculs spécialisées. L'idée d'en ajuster non pas un, mais jusqu'à six simultanément (nombre de sommets de simplexes typiquement utilisé dans la méthode des sous-espaces), semble donc peu envisageable. En revanche, les méthodes d'ajustement par prompts continus et, par extension, les méthodes PEFT, proposent non pas d'ajuster ces modèles de langue directement, mais au contraire d'y introduire de nouveaux paramètres apprenables (à l'instar des embeddings des tokens virtuels en apprentissage de prompts continus), et d'ajuster ceux-ci tout en figeant les paramètres pré-entraînés du modèle de langue. L'avantage principal de cette approche réside dans la capacité de ces modèles "adaptés" à répliquer (voire améliorer dans des contextes associés à des tailles d'échantillons faibles) les performances des modèles de langue tout en réduisant leur nombre de paramètres apprenables de plusieurs ordres de grandeur. Ces approches nécessitent de plus typiquement une initialisation aléatoire des paramètres additionnels qu'elles introduisent dans le modèle, en faisant ainsi des candidats naturels particulièrement prometteurs pour l'adaptation de la méthode des sous-espaces aux gros modèles de langage.

Les contributions de cet article sont les suivantes. Tout d'abord, nous introduisons la première adaptation de la méthode des sous-espace aux gros modèles de langage, via d'ajustement de sous-espace de préfixes (une méthodes PEFT similaire à l'ajustement de prompts continus parmi les plus performantes dans la littérature académique actuelles). Ensuite, cet article propose d'exploiter certains avantages naturels que la méthode des sous-espaces offre afin de revisiter la notion de guidage d'ajustement d'un modèle par la métrique de validation. On montrera empiriquement que la combinaison de ces deux idées amène un gain conséquent en termes de prédiction moyenne sur les tâches de compréhension du langage naturel que propose le benchmark GLUE (Wang *et al.*, 2018).

Finalement, une étude d'ablation sera présentée pour fournir quelques éléments d'explication quant aux mécanismes permettant ce gain en termes de prédiction.

2 Méthode des sous-espaces

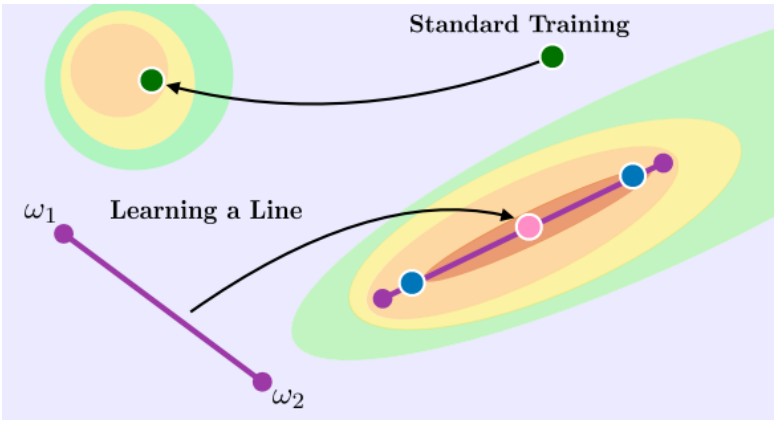


FIGURE 1 – Illustration de la méthode d'apprentissage de sous-espaces. On cherche à obtenir non pas un unique modèle, ici représenté comme un point sur la surface de la fonction objectif, mais une droite entière associée à de faibles valeurs de cette surface. Illustration tirée de (Wortsman *et al.*, 2021)

La méthode d'apprentissage de sous-espaces de réseaux neuronaux, illustrée en figure 1, permet d'obtenir en un seul processus d'ajustement une région connexe de l'espace des paramètres composée de modèles autant divers que tous associés à des performances. On choisi de définir ce domaine de modèles Λ comme un simplexe, caractérisé par ses m sommets $\{\omega_i\}_{i=1}^m$ comme l'ensemble des barycentres de ces derniers :

$$P(\alpha, \{\omega_i\}_{i=1}^m) = \Sigma_{i=1}^m \alpha_i \omega_i \; avec \; \{\alpha \in \mathbb{R}^m : \Sigma_i \alpha_i = 1, \alpha_i > 0\} \tag{1}$$

L'objectif est donc de minimiser la fonction coût choisie l pour tout paramétrage de modèle appartenant à Λ. Autrement dit, on cherche à minimiser l'espérance de l mesurée sur la distribution des données D pour tout modèle f échantillonné uniformément du simplexe Λ paramétré par α :

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbb{E}_{\alpha\sim U(\Lambda)}[l(f(\mathbf{x}, P(\alpha, \{\omega_i\}_{i=1}^m)), \mathbf{y})]] \tag{2}$$

Avec :

- D la distribution des données,
- $U(\Lambda)$ la distribution uniforme sur le domaine Λ,
- $(\mathbf{x}, \mathbf{y})$ les variables d'entrées et la variable à expliquer, respectivement,
- f un modèle prédictif paramétré par un élément quelconque de D.

En pratique, l'ajustement du simplexe de modèle ne se fait pas par optimisation directe de cette entité,

mais d'une approximation stochastique échantillonnée à la fois au niveau des données (comme en descente stochastique traditionnelle) et de $U(D)$. En d'autres termes, l'initialisation du simplexe est tout d'abord typiquement réalisée via l'initialisation indépendante de chacun de ses sommets, par le biais de méthodes traditionnelles d'initialisation aléatoire de modèles neuronaux.

Par la suite, et ce pour chaque itération de descente de gradient, un modèle paramétré en θ est choisi par échantillonnage uniforme sur le simplexe, et utilisé dans le cadre d'une descente de gradient stochastique traditionnelle. Le gradient de la fonction objectif peut ensuite être propagé aux sommets du simplexe en écrivant :

$$\frac{\partial l}{\partial \omega_i} = \frac{\partial l}{\partial \theta} \frac{\partial P(\alpha, \{\omega_i\}_{i=1}^m)}{\partial \theta} \tag{3}$$

Dans le but de garantir une certaine diversité fonctionnelle au sein du simplexe de modèles, un terme de régularisation est ajouté à l'objectif, encourageant la dissimilarité (au sens de la similitude cosine) entre les différents sommets du simplexe :

$$\beta.\mathbb{E}_{j\neq k}[cos^2(\omega_j, \omega_k)] = \beta.\mathbb{E}_{j\neq k}[\frac{\langle \omega_j, \omega_k \rangle^2}{\|\omega_j\|_2^2 \|\omega_k\|_2^2}] \tag{4}$$

L'intensité β de ce terme de régularisation constitue un hyperparamètre au modèle, qu'on fixe à la valeur recommandée par défaut de $\beta = 1$ (Wortsman *et al.*, 2021).

On dispose donc, après descente de gradient, d'un simplexe entier de modèles que l'on peut échantillonner à volonté pour inférence. Le centroïde du simplexe, en particulier, présente typiquement des capacités de généralisation supérieures à celles de modèles obtenus par ajustement traditionnel.

Une possible justification de cette propriété, visualisée en figure 2, réside dans l'idée qu'un modèle obtenu par ajustement traditionnel serait localisé à la périphérie d'un minimum local de la fonction objectif, typiquement plus sensible à des erreurs de généralisation (Izmailov *et al.*, 2018; Dziugaite & Roy, 2018). Parcourir le sous-espace permet au contraire de "traverser" le minimum local, afin d'obtenir un modèle associé à une zone plus stable de la fonction objectif.

Jusqu'ici, la définition de la procédure d'ajustement ne dépend aucunement de l'aspect connectionniste des réseaux neuronaux, et considère simplement un modèle comme un vecteur de paramètres apprenables. L'immense majorité des modèles d'apprentissage profond, en revanche, sont définis par une succession de transformations non linéaires. Il semble donc naturel d'incorporer, d'une manière ou d'une autre, cette structure séquentielle des modèles neuronaux dans la définition de la procédure d'ajustement. Pour ce faire, il est conseillé d'échantillonner les paramètres de chaque couche du modèle indépendamment. Cette variante dite "couche par couche" de la méthode est typiquement associée à de meilleures performances prédictives.

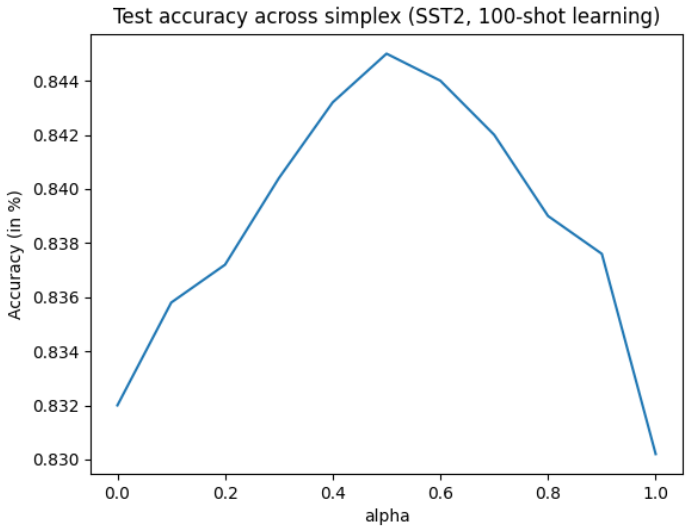


FIGURE 2 – Évolution des performances prédictives d'un modèle de langue ajusté sur une droite de préfixes. Les performances de généralisation suivent une courbe similaire à une parabole maximale en son centre

3 Méthode proposée

3.1 Apprentissage de sous-espaces de préfixes

Comme mentionné en introduction, la méthode des sous-espaces est en pratique difficilement applicable dans le cadre de l'ajustement classique de modèles de langue. Les méthodes de prompts, en revanche, reposent sur l'apprentissage non pas du modèle lui-même, mais sur l'ajustement de n vecteurs d'embeddings $\{E_i\}_{i=1}^{n}$ concaténés typiquement au début de la séquence d'embeddings d'entrée du modèle de langue LM_Φ paramétré en Φ. En d'autres termes, pour une séquence d'entrées de L tokens, $\{I_i\}_{i=1}^{l}$ on construit un modèle prédictif à partir non pas de la sortie du modèle de langage :

$$LM_\Phi(\{I_i\}_{i=1}^{l}) \tag{5}$$

mais de

$$LM_\Phi(concat(\{E_i\}_{i=1}^{n}, \{I_i\}_{i=1}^{l})) \tag{6}$$

L'ajustement du modèle prédictif se fait uniquement par ajustement des tokens virtuels $(E_i)_{i=1}^{n}$, tout en figeant les paramètres du modèle de langue Φ.

Afin d'augmenter l'expressivité de cette méthode (particulièrement limitée en terme de nombre de paramètres apprenables), la méthode d'apprentissage de préfixes (Li & Liang, 2021), choisie dans cet article comme candidat à l'application de la méthode des sous-espaces, propose de concaténer ces tokens virtuels non pas à la séquence d'entrée du modèle, mais aux séquences Key $\{K_i\}_{i=1}^{l}$ et Values $\{V_i\}_{i=1}^{l}$ utilisées en entrée des modules d'attention multiplicative présents dans chaque couche du modèle de langue. En d'autre terme, la sortie $\{H_i\}_{i=1}^{l}$ d'un module d'attention Att d'une couche de modèle de langage, initialement définie comme :

$$\{H_i\}_{i=1}^{l} = Att_{\Psi}(\{K_i\}_{i=1}^{l}; \{Q_i\}_{i=1}^{l}; \{V_i\}_{i=1}^{l}) \tag{7}$$

est définie dans le cadre de l'apprentissage de préfixes comme :

$$\{H_i\}_{i=1}^{l} = Att_{\Psi}(concat(\{E_i^k\}_{i=1}^{n}, \{K_i\}_{i=1}^{l}); \{Q_i\}_{i=1}^{l}; concat(\{E_i^v\}_{i=1}^{n}, \{V_i\}_{i=1}^{l})) \tag{8}$$

Avec :

- $\{H_i\}_{i=1}^{l}$ la sortie du module d'attention
- Att le module d'attention multiplicative d'une couche de modèle de langage
- $\{K_i\}_{i=1}^{l}$ la séquences Keys d'entrées au module d'attention (obtenue par transformation linéaire de la séquence d'entrée de la couche du modèle de langue)
- $\{V_i\}_{i=1}^{l}$ la séquences Values d'entrées au module d'attention (obtenue par transformation linéaire de la séquence d'entrée de la couche du modèle de langue)
- $\{K_i\}_{i=1}^{l}$ la séquences Queries d'entrées au module d'attention (obtenue par transformation linéaire de la séquence d'entrée de la couche du modèle de langue)
- $\{E_i^k\}_{i=1}^{n}$ les vecteurs d'embeddings apprenables concaténé à la séquence Keys
- $\{E_i^v\}_{i=1}^{n}$ les vecteurs d'embeddings apprenables concaténé à la séquence Values

Dans une approche similaire à la méthode de prompts, l'ajustement de préfixes se fait uniquement par ajustement (via descente de gradient) des tokens virtuels $\{E_i^k\}_{i=1}^{n}$ et $\{E_i^v\}_{i=1}^{n}$, tout en laissant les paramètres du modèle de langage lui même figés. Apprendre directement ces embeddings s'avère en revanche particulièrement instable. Aussi, il est coutume de pas les ajuster directement, mais d'utiliser une astuce de reparamétrisation, consistant à concaténer aux séquences Keys et Values non pas les séquences de préfixes directement, mais une transformation de ces derniers, paramétrée par un perceptron sous complets à deux couches, comme illustré en figure 3-1.

L'adaptation de la méthode des sous-espaces à l'ajustement de préfixes peut donc se faire par deux approches distinctes :

1. Application aux paramètres apprenables du modèle eux même, et donc à l'embedding initial et au perceptron sous-complet de reparamétrisation
2. Application de la méthode aux préfixes eux même, et donc à la sortie du module de reparamétrisation

On propose dans cet article d'étudier l'option 2 la méthode des sous-espaces, et donc d'appliquer la méthode de sous-espaces directement aux préfixes $\mathbf{E}^k$ et $\mathbf{E}^v$, comme illustré en figure 3.2.

Il convient également de s'intéresser à l'adaptation de la variante "couche par couche" de la méthode. En effet, l'ajustement de préfixe ne repose pas sur l'introduction dans le modèle de langue d'une structure classique de perceptron, mais d'une modification de l'opération du module d'attention multitête de ce dernier. Nous proposons dans cet article d'étendre cette variante "couche par couche" à $\mathbf{E}^k$ et $\mathbf{E}^v$. Ainsi, à chaque itération de descente durant l'ajustement, les $\mathbf{E}^k$ et $\mathbf{E}^v$ de chaque couches seront tous échantillonnés indépendamment. De plus, cet échantillonnage sera effectué indépendamment pour toutes les observations, contrairement à l'approche traditionnelle qui préfère créer un unique modèle par itération de descente.

Additionnellement, la tête de prédiction du modèle est typiquement initialisée aléatoirement. On choisit donc de lui appliquer également la méthode des sous-espaces, comme décrite en partie 2. Par

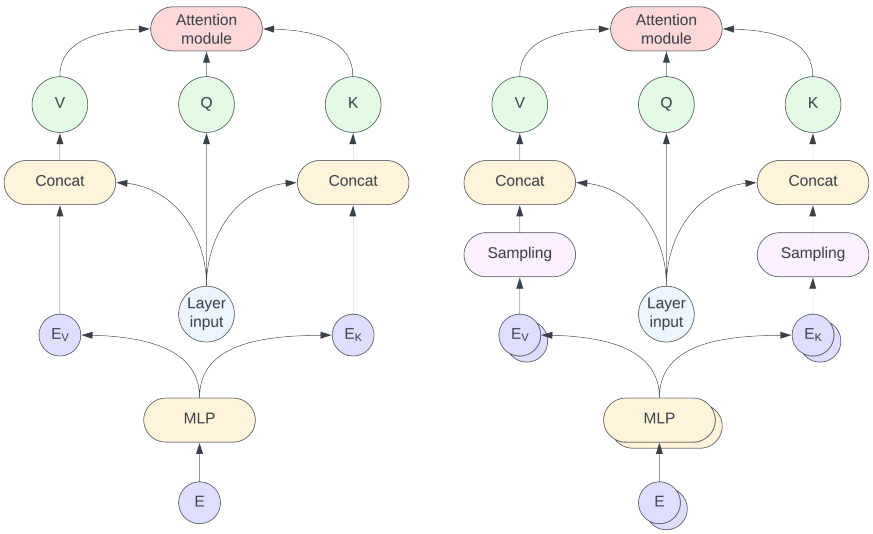


FIGURE 3 – 3.1 (gauche) : Méthode d'apprentissage de préfixe. Un premier embedding est utilisée comme entrée d'un perceptron à deux couches, dont la sortie est concaténée aux séquences de Keys et Values des modules d'attention multi-tête du modèle de langue. Les paramètre apprenables du modèles sont l'embedding initial et le perceptron de reparamétrisation. 3.2 (droite) : Proposition d'extension à l'apprentissage de sous-espaces de préfixes. Chaque sommet du simplexe est calculé indépendamment, et l'échantillonnage aléatoire est effectué dans le simplexe de préfixes (par opposition au simplexe de l'embedding et du perceptron de reparamétrisation)

souci de cohérence, la variante d'échantillonnage de paramètres à l'échelle de l'observation, et non de la batch, lui sera en revanche également appliquée.

En résumé, les paramètres ajustables du modèle prédictif dans un cadre d'apprentissage de préfixe sont les suivant :

En résumé, nous proposons d'ajuster un simplexe à m sommets de préfixes de la manière suivante :

- Initialisation indépendante de m systèmes de reparamétrisation
- Calcul, pour chaque itération de descente, des m sommets du simplexe
- Construction des préfixes utilisés pour l'inférence de la fonction coût et le calcul de son gradient par échantillonnage uniforme indépendamment pour chaque observation, chaque couche, ainsi que pour les préfixes des séquences Keys et Values

3.2 Méthode des sous-espaces et inférence stochastique de métriques de validation

L'ajustement d'un gros modèle de langue est typiquement guidé par l'estimation d'une métrique de performance sur un jeu de validation, ceci autant durant la recherche d'hyperparamètres, que durant le processus de descente même, où le meilleur modèle au sens de cette valeur scalaire est retenue comme modèle final. La question de l'estimation de cette métrique dans un cadre d'apprentissage de sous-espace soulève des questions. En effet, l'ajustement non pas d'un seul modèle, mais tout un

simplexe, résulte en autant de métriques de validation potentiellement estimables.

Puisque l'on se limite dans le cadre de cet article à l'utilisation de cette méthode afin d'en extraire le centroïde -associé à de meilleures performances de généralisation- il serait naturel de l'estimer vis-à-vis de ce dernier. Cependant, l'existence pour l'inférence non pas d'un unique modèle, mais de ce simplexe, et notamment des informations supplémentaires qu'il apporte quant à la nature du minimum local obtenu, peut s'avérer intéressante. C'est notamment le cas dans un contexte de *few-shot learning*. En effet, comme mentionné précédemment, pour des jeux de données de validation associés à des tailles d'échantillon faibles (< 100 typiquement), l'estimation de cette métrique peut devenir déraisonnablement bruité. Cette propriété a deux effets indésirables sur le processus d'élaboration du modèle :

— La recherche d'hyperparamètres perd sensiblement de son intérêt, puisque la valeur guidant la sélection de modèles n'est plus informative

— Plus important encore, il devient particulièrement difficile durant l'ajustement d'identifier le surajustement de modèle. Les méthodes d'arrêt précoces, pourtant particulièrement importante dans un cadre de *few-shot learning* (par exemple pour limiter le surajustement), perdent donc également en efficacité

On propose donc d'utiliser le simplexe dans son ensemble pour "augmenter" l'estimation de la métrique de validation. Celle-ci se fera donc non pas en utilisant le centroïde du simplexe, mais avec *plusieurs* modèles échantillonnés aléatoirement pour chaque individu du jeu de validation. En d'autre termes, on présentera à chaque estimation de métrique le jeu de validation n fois, et le processus d'inférence se fera dans les mêmes conditions que pendant l'ajustement, c'est-à-dire avec échantillonnage aléatoire du modèle utilisé pour chaque observation. On fixe le nombre n de répétitions du jeu de données de validation à 10 dans toutes les expériences présentées dans cet article qui utilise cette approche d'inférence stochastique.

On sélectionne tout de même en guise de modèle final le centroïde du simplexe. En effet, le déterminisme d'un modèle reste une propriété désirable dans des situations de production.

4 Expériences

Toutes les expériences décrites dans cet article afin d'évaluer les performances prédictives de la méthode proposée sont réalisées avec BERT-base-cased sur des jeux de données construits à partir du benchmark GLUE (Wang *et al.*, 2018), un corpus de 8 tâches de compréhension du langage anglais, toutes formulées comme des problèmes de classification. Cependant, de part l'inaccessibilité des jeu de test sur ce benchmark ainsi que leur taille d'échantillon sensiblement trop importante pour être pertinente dans un cadre de "few-shot learning", nous n'utilisons pas directement ces jeux de données, mais en construisons de nouveaux plus adaptés à notre problématique suivant une méthodologie similaire à celle présentée dans (Mao *et al.*, 2022). Nous construisons par échantillonnage aléatoire des séries de jeux de données de tailles d'échantillon variable (50, 100, 200 et 500 observations). Notre méthode de construction de ces jeux de données diffère cependant de la leur sur quelques points clés.

Premièrement, les auteurs ont choisi de construire des jeux de validation de 1 000 d'observations pour tous leurs jeux de données d'apprentissage, ce qui, de notre avis, n'est pas réaliste dans un contexte de "few-shot learning" (un jeu de validation ne contient généralement pas dix fois plus d'exemple que son jeu d'entraînement). Deuxièmement, ils utilisent les jeux de validation du benchmark GLUE comme

jeux de test. Toutefois, certains de jeux validation ont des tailles d'échantillon faibles (277 pour RTE, 408 pour MRPC), pouvant potentiellement bruiter l'estimation des métriques de performances. Par conséquent, pour chaque jeu de données de référence, un ensemble de données de taille d'échantillon K est construit comme suit :

— Les jeux d'entraînement et de validation sont concaténés en un unique jeu
— La moitié des observations (plafonnée à 5000 observations) sont exclues de ce jeu de données pour construire un jeu de données de test, commun à toutes les expériences
— K observations sont ensuites sélectionnées par échantillonnage uniforme, et répartie dans un jeu d'entraînement et de validation suivant des proportions 70/30

Pour chaque tâche du benchmark, et pour chacune des tailles d'échantillon retenues, 10 jeux de données sont construits en suivant cette méthodologie, afin de permettre de répliquer les expériences sur différents jeux de données, de permettre l'estimation de performances moyennes, et de tester la significativité au seuil des 5% des différences obtenues (via bootstrap).

Pour tous ces jeux de données, nous comparons notre méthode à 5 modèles d'ajustement de référence, dont 4 méthodes PEFT. Ces méthodes, de manière similaires à la méthode d'apprentissage de préfixes, se basent sur l'idée de figer les paramètres du modèle de langue, et d'introduire une fraction de nouveaux paramètres ajustables (typiquement d'une cardinalité inférieure de plusieurs ordres de grandeur à celle du modèle lui même), mais elles diffèrent dans la manière dont elles introduisent ces nouveaux paramètres dans le modèle :

— Par ajustement traditionnel (Devlin *et al.*, 2019), où l'intégralité des paramètres du modèle de langue sont ajustés par descente de gradient. Cette méthode reste parmi les plus usitées en traitement automatique des langues, et représente donc une référence essentielle à laquelle comparer la méthode que nous proposons
— Par adapteur standard (Houlsby *et al.*, 2019), qui proposent typiquement d'introduire un ou plusieurs perceptron à deux couches "bottleneck" à différents étages d'une couche de Transformer, première méthode PEFT a avoir été introduite, et la plus reconnue
— Par Low Rank Adaption (LORA) (Hu *et al.*, 2021), qui reparamétrise les matrices de projections des Values et Queries précédent le module d'attention multiplicative multitête par via deux perceptrons sous complets linéaires à deux couches, première méthode PEFT à proposer différentes transformations pour différents éléments du module d'attention
— Par méthode UniPELT (Mao *et al.*, 2022) méthode de fusion combinant adapteurs, LoRA et préfixes afin de bénéficier des avantages de chacunes (sans souffrir de leurs potentiels inconvénients respectifs)
— Par apprentissage de préfixes classique, méthode de référence cruciale afin d'estimer la part des performances de la méthode proposée qui lui est réellement attribuable

Pour toutes les méthodes, nous suivons la même procédure d'apprentissage et de recherche d'hyperparamètres que proposée par (Mao *et al.*, 2022). Tous les modèles ont été ajustés pendant 50 epochs en utilisant les réglages d'usine du Trainer Huggingface, ainsi qu'un mécanisme d'arrêt anticipé munie d'une patience de 10 epochs. La taille de batch est fixée à 16 pour toutes les expériences, et les recherches d'hyperparamètres sont effectuées par recherche exhaustive dans les valeurs suivantes :

— Ajustement traditionnel : Taux de descente parmi $[1e-5, 2e-5]$
— Adapteur standard : Taux de descente de $1e-4$ et taux de réduction parmi $[3, 6, 12]$
— LoRA : Rang et valeur alpha fixés à 8, taux de descente parmi $[1e-4, 5e-4]$
— UniPELT : Longueur de préfixe fixée à 10, adapteur à taux de réduction fixé à 16, et LoRA

avec range et valeur alpha fixés à 8. Taux de descente parmi $[2e-4, 5e-4]$

— Apprentissage de préfixe : Longueur de préfixe fixée à 50, taux de descente parmi $[1e-4, 2e-4, 5e-4]$

Pour garantir une comparabilité optimale, le choix des hyperparamètres de la méthode proposée seront choisis de manière à correspondre exactement à ceux de la baseline d'apprentissage par préfixe, eux aussi déterminés pour la première fois dans un cadre de classification de texte par (Mao *et al.*, 2022). Les sous-espaces ajustés dans les expériences sont tous des simplexes à 6 sommets.

5 Résultats

Method	MNLI	QNLI	SST-2	QQP	CoLA	STS-B	MRPC	RTE	Avg.
[*K = 50]*									
Ajustement classique	35.5	65.9	57.57$_*$	45.6$_*$	**3.5**	45.1$_*$	81.1	50.6	48.1
Adapteur	35.6	62.6$_*$	64.7$_*$	35.3$_*$	0.0	59.7	80.2	**53.1***	48.9
LoRA	35.7	63.9$_*$	68.4$_*$	47$_*$	1.0	56.5$_*$	**81.4**	52.8	50.8
UniPELT	35.3	62.4$_*$	73.1$_*$	42.3$_*$	1.1	**64.3***	80.7*	51.8	51.4
Préfixes	**37.8**	63.5$_*$	74.9$_*$	53.1	1.8	59.2$_*$	80.4*	52.6	52.9
Préfixes (sous-espace)	36.6	**66.6**	**80.1**	**54.3**	0.8	61.1	80.0	52.2	**54.0**
[*K = 100]*									
Ajustement classique	35.5$_*$	68.9$_*$	73.9$_*$	52.6$_*$	3.0$_*$	64.1$_*$	81.3	52.1	53.9
Adapteur	36.3$_*$	66.7$_*$	72.8$_*$	54.0$_*$	7.2	63.8$_*$	80.5	53.0	54.3
LoRA	37.3	64.9$_*$	73.2$_*$	54.2$_*$	7.3	60.4$_*$	81.3	52.9	53.9
UniPELT	37.7	66.9$_*$	79.1$_*$	53.6$_*$	5.1	68.4	79.7$_*$	52.0$_*$	55.3
Préfixes	38.3	69.4$_*$	80.8$_*$	57.2	**8.1**	66.6$_*$	81.1	**54.2**	57.0
Préfixes (sous-espace)	**38.5**	**70.8**	**82.5**	**59.6**	7.8	**68.3**	**81.5**	54.1	**57.9**
[*K = 200]*									
Ajustement classique	42.3	**71.9**	80.8$_*$	63.0	20.2	69.0$_*$	80.8	54.6	60.3
Adapteur	42.7	69.1$_*$	83.1$_*$	59.5$_*$	**26.5***	70.3$_*$	80.7	**56.2**	61.0
LoRA	41.0	67.1$_*$	82.2$_*$	61.2$_*$	19.8	67.8$_*$	80.1	54.5	59.2
UniPELT	41.6	70.2	82.8$_*$	58.7$_*$	16.4	**72.8**	**81.7**	54.9	59.9
Préfixes	**44.9**	71.4	**84.2**	63.0$_*$	22.2	71.3	79.6$_*$	56.0	61.6
Préfixes (sous-espace)	44.7	71.2	84.1	**64.4**	21.1	72.3	81.6	**55.9**	**61.9**
[*K = 500]*									
Ajustement classique	52.7$_*$	74.3$_*$	85.4$_*$	66.8	32.2$_*$	78.0	82.5	59.8	66.5
Adapteur	51.1$_*$	72.4$_*$	85.4$_*$	65.7$_*$	**38.9***	76.1$_*$	81.9$_*$	59.8	66.4
LoRA	50.1$_*$	73.6$_*$	84.6$_*$	66.5	35.3	75.6$_*$	82.3$_*$	58.3$_*$	65.8
UniPELT	50.7$_*$	74.2$_*$	85.4$_*$	63.4$_*$	34.2	77.2	82.1	57.8$_*$	65.6
Préfixes	54.0$_*$	74.7$_*$	85.6$_*$	66.2	35.7	77.8	82$_*$	60	67.0
Préfixes (sous-espace)	**55.7**	**75.4**	**86.1**	**67.2**	36.0	**78.1**	**83.1**	**60.8**	**67.8**

TABLE 1 – Résultats de l'expérience. Des mesure F1 sont reportées pour QQP et MRPC. Une corrélation de Spearman est reportée pour STS-B. Une corrélation de Matthews pour CoLA. Des mesures d'accuracy sont reportées pour le reste des tâches. Les résultats en gras et soulignés correspondent aux premières et secondes meilleures performances, respectivement. Les résultats suivis d'une astérisque en indice ou exposant correspondent à des résultats significativement supérieurs ou inférieurs à ceux de la méthode proposée, respectivement

Les performances de toutes les méthodes PEFT sélectionnées ainsi que de l'approche proposée sont

présentées dans la Table 1, ceci pour toutes les différentes tâches du benchmark GLUE, et pour les différentes tailles d'échantillons retenues. Dans l'ensemble, l'apprentissage de sous-espace de préfixes surpasse toutes les autres méthodes de base en moyenne, ceci pour toutes les tailles d'échantillons. La méthode présente notamment un gain d'un point en comparaison à l'apprentissage de préfixes classique, pour des tailles d'échantillons de 50 et 100 observations. Ce gain diminue mais reste présent lorsque la taille des jeux de données échantillonnés augmente, ce qui n'est pas nécessairement surprenant, l'apprentissage de sous-espace améliorant principalement la capacité de généralisation du modèle final.

La comparaison entre la méthode par apprentissage de sous-espace de préfixes surpasse l'apprentissage de préfixes traditionnel est particulièrement intéressante. En effet, les deux approches reposent essentiellement sur le même formalisme. En terme de significativité statistique, la méthode proposée surpasse son équivalent classique 12 fois :

— Sur QNLI, SST-2, et STS-B pour $K = 50$ et $K = 100$
— Sur MRPC et QQP pour $K = 200$
— Sur MNLI, QNLI, MRPC et STS-B pour $K = 500$

Elle est en revanche surpassée statistiquement une unique fois, sur MRPC pour $K = 50$, ce qui est d'autant plus surprenant quand on remarque que la différence entre les deux méthodes sur cette expérience est de 0.4%. De plus, la méthode proposée redevient significativement supérieure sur cette tâche une fois que la taille d'échantillon augmente jusqu'à 500 observations.

Plus largement, la méthode proposée n'est surpassé significativement que 6 fois sur toutes les expériences :

— Sur MRPC par les méthodes de préfixe et LoRA pour $K = 50$
— Sur RTE par la méthode d'Adapteur pour $K = 50$
— Sur STS-B par la méthode UniPELT pour $K = 50$
— Sur CoLA par la méthode d'Adapteur pour $K = 200$ et $K = 500$

On remarquera notamment que tous ces événements s'observent pour $K = 50$ (et donc de jeux de validations de 15 observations), où l'ajustement de modèles devient particulièrement complexes.

La méthode proposée surpasse en revanche significativement l'une des autres méthodes de base à travers les expériences effectuées un total de 80 fois, montrant un clair avantage en termes de pouvoir prédictif.

On remarque notamment que la majorité des expériences où la méthode proposée surpasse les méthodes de référence principalement sur 3 jeux de données, à savoir QNLI, SST-2 et QQP. De plus, la capacité de la méthode proposée à surpasser significativement les méthodes de référence sur ces tâches ne semble pas dépendre de la taille d'échantillon des jeux de données.

Il est en revanche difficile d'identifier ce qui différencie ces jeux de données de ceux où la méthode proposée reste comparable aux méthodes de références. En effet, les deux camps présentent autant de tâches similaires, et autant de jeux de données déséquilibrés.

6 Étude d'ablation

De manière à mieux identifier l'impact des différents aspects de la méthode proposées, une étude d'ablation est également rapportée en table 2, avec les variantes suivantes :

— Même méthode avec des simplexes à 2 sommets (ie une ligne)

— Même méthode sans l'inférence de validation stochastique
— Même méthode sans sous-espaces des têtes de prédiction
— Même méthode sans sous-espace de préfixe (donc uniquement sur les têtes de prédictions)

Method	$K = 50$	$K = 100$	$K = 200$	$K = 500$
Méthode proposée	54.0	57.9	61.9	67.8
Simplexe à 2 sommets	53.6	57.9	62.1	67.6
Validation déterministe	49.5	56.0	61.4	67.8
Sans sous-espace de tête	53.5	56.8	61.3	67.2
Sans sous-espace de préfixes	52.6	57.7	62.2	67.6

TABLE 2 – Résultats de l'étude d'ablation. Les scores rapportés correspondent à la moyenne des performances prédictives sur toutes les tâche du benchmark GLUE

Les résultats de cette étude d'ablation peuvent être résumés comme suit :

1. L'utilisation de simplexes à deux sommets montrent des performances légèrement inférieures à la méthode proposées pour $K = 50$, puis des performances similaires par la suite
2. L'utilisation de la méthode de sous-espace guidée par une métrique de validation estimée de manière déterministe s'effondre pour $K = 50$, $K = 100$ et $K = 200$ (cas pour lesquels les performances sont d'ailleurs inférieures à la méthode d'ajustement de préfixes classique), puis finissent par devenir équivalents à la méthode proposée
3. L'utilisation de sous-espaces de préfixes, sans sous-espace de tête, est surpassée avec consistance par la méthode proposée
4. L'utilisation de sous-espace de tête de prédiction, couplée à des préfixes classique, est surpassée de manière considérable pour $K = 50$ (cas où cette approche est moins performante que la méthode d'ajustement de préfixes classique), et similaire à la méthode proposée lorsque la taille d'échantillon augmente

Ces observations, prises dans leur ensemble, amènent notamment plusieurs éléments de preuves quant à la pertinence de l'utilisation en "few-shot learning" de la notion que nous proposons d'inférence stochastique de métrique de validation. L'observation 3, en particulier, montre que l'ajustement de sous-espace de préfixes avec estimation classique de la métrique de validation n'est associé aux mêmes gains de performances que la méthode proposée *qu'à partir* de $K = 500$. Que les performances de cette variante soient de plus inférieures à celles obtenues par ajustement de préfixes classiques vient encore appuyer l'importance de la méthode proposée d'inférence stochastique de métrique de validation.

Les observation 1, 2, 3 permettent par la suite d'amener des arguments légèrement plus faibles sur l'importance de la taille du simplexe dans le cadre de cette estimation stochastique. En effet, bien que la taille du simplexe ne semble pas avoir d'effet pour $K > 50$ (indiquant fortement qu'il est préférable d'apprendre des lignes pour ces tailles d'échantillons, considérablement plus économes en terme de complexité mémoire), elle semble en avoir un pour des tailles d'échantillon très faible. Cette observation pourrait s'expliquer par la richesse de l'information extraites du jeu de données de validation par estimation stochastique, dûe à un simplexe plus important. Les résultats présentés dans cet article sont en revanche insuffisants pour confirmer (ou infirmer) cette hypothèse.

Similairement, les observations 2 et 3 montrent tout particulièrement l'importance d'ajuster par sous-espace l'intégralité des paramètres apprenables du modèle dans le cas où $K = 50$. Ceci pourrait également s'expliquer en avançant l'idée que l'on perd en capacité à caractériser le minimum local obtenu en limitant l'estimation stochastique de la métrique de validation à une sous partie des paramètres apprenables du modèle.

7 Conclusion

On a dans cet article introduit deux idées novatrices. La première, une adaptation de la méthode des sous-espaces à l'ajustement de gros modèles de langues par le biais de méthode PEFT, est à notre connaissance le premier exemple d'utilisation de cette méthode dans la littérature académique portant sur le traitement automatique des langues. La seconde, proposant une manière alternative d'estimation des métriques de validation, constitue une application originale de la méthode des sous-espaces et n'est en aucun cas spécifique à des problématiques rencontrées en analyse de données textuelles. L'utilisation jointe de ces deux méthodes donne lieu à une augmentation considérables des performances de modèles de langues communs comme BERT, sur les tâches de compréhension du langage proposées par le benchmark GLUE reformulées dans un contexte de "few-shot learning". L'étude d'ablation présentée en fin d'article permet en outre de poser des hypothèses quant à l'impact de ces deux contributions. Le gain de performances observés sur très petits jeux de données (≤ 100) semble être en effet principalement expliqué par l'information plus fine extraite du jeu de validation via la méthode d'estimation stochastique de métrique. Ce gain semble en revanche se dissiper pour des tailles d'échantillon plus élevées, ou la méthode des sous-espaces appliquées à l'apprentissage de préfixes semble se suffire à elle-même pour permettre un gain de performance sur les méthodes PEFT, ainsi que sur l'ajustement de modèle classique.

L'application de la méthode des sous-espaces aux méthodes PEFT permet en outre l'ajustement de puissants modèles prédictifs tout en réduisant considérablement les ressources machines typiquement nécessaires à l'ajustement des gros modèles de langues, ne dénaturant pas ainsi la volonté fondamentale d'*efficience* de ces méthodes. L'apprentissage de sous-espaces de préfixes reste ainsi accessible même dans des situations où les ressources, autant en données qu'en puissance de calcul, sont limitées.

Remerciements

Nous tenons à remercier le Sorbonne Center for Artificial Intelligence pour le financement du contrat post-doctoral de Louis Falissard au sein du laboratoire MLIA de l'Institut des Systèmes Intelligents et de Robotique.

Références

BAPNA A. & FIRAT O. (2019). Simple, scalable adaptation for neural machine translation. 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. 1538–1548, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1165.

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.

DZIUGAITE G. K. & ROY D. (2018). Entropy-SGD optimizes the prior of a PAC-Bayes bound : Generalization properties of entropy-SGD and data-dependent priors. In J. DY & A. KRAUSE, Éds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, p. 1377–1386 : PMLR.

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.

HU E. J., SHEN Y., WALLIS P., ALLEN-ZHU Z., LI Y., WANG S., WANG L. & CHEN W. (2021). Lora : Low-rank adaptation of large language models. DOI : 10.48550/ARXIV.2106.09685.

IZMAILOV P., PODOPRIKHIN D., GARIPOV T., VETROV D. & WILSON A. G. (2018). Averaging weights leads to wider optima and better generalization. DOI : 10.48550/ARXIV.1803.05407.

LI X. L. & LIANG P. (2021). Prefix-tuning : Optimizing continuous prompts for generation. 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. 4582–4597, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.353.

MAO Y., MATHIAS L., HOU R., ALMAHAIRI A., MA H., HAN J., YIH S. & KHABSA M. (2022). UniPELT : A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 6253–6264, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.433.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language Models are Unsupervised Multitask Learners.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. DOI : 10.48550/ARXIV.1910.10683.

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.

WORTSMAN M., HORTON M. C., GUESTRIN C., FARHADI A. & RASTEGARI M. (2021). Learning neural network subspaces. In M. MEILA & T. ZHANG, Éds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 de *Proceedings of Machine Learning Research*, p. 11217–11227 : PMLR.

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

Recherche cross-modale pour répondre à des questions visuelles

Paul Lerner[1] Olivier Ferret[2] Camille Guinaudeau[3]
(1) Université Paris-Saclay, CNRS, LISN, 91400, Orsay, France
(2) Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
(3) Université Paris-Saclay, CNRS, JFLI, 101-0003, Tokyo, Japon
prenom.nom@lisn.upsaclay.fr, prenom.nom@cea.fr

RÉSUMÉ

Répondre à des questions visuelles à propos d'entités nommées (KVQAE) est une tâche difficile qui demande de rechercher des informations dans une base de connaissances multimodale. Nous étudions ici comment traiter cette tâche avec une recherche cross-modale et sa combinaison avec une recherche mono-modale, en se focalisant sur le modèle CLIP, un modèle multimodal entraîné sur des images appareillées à leur légende textuelle. Nos résultats démontrent la supériorité de la recherche cross-modale, mais aussi la complémentarité des deux, qui peuvent être combinées facilement. Nous étudions également différentes manières d'ajuster CLIP et trouvons que l'optimisation cross-modale est la meilleure solution, étant en adéquation avec son pré-entraînement. Notre méthode surpasse les approches précédentes, tout en étant plus simple et moins coûteuse. Ces gains de performance sont étudiés intrinsèquement selon la pertinence des résultats de la recherche et extrinsèquement selon l'exactitude de la réponse extraite par un module externe. Nous discutons des différences entre ces métriques et de ses implications pour l'évaluation de la KVQAE.

ABSTRACT

Cross-modal retrieval for Knowledge-based Visual Question Answering

Knowledge-based Visual Question Answering about named Entities (KVQAE) is a challenging task that requires retrieving information from a multimodal Knowledge Base. To tackle this task, we study cross-modal retrieval and its combination with mono-modal retrieval. We focus on the CLIP model, a multimodal model trained on images paired with their textual caption. We show that cross-modal outperforms mono-modal retrieval but also that the two are complementary and can be easily combined. We show that cross- outperforms mono-modal retrieval but also that the two are complementary and can be easily combined. We also study different fine-tuning strategies for CLIP and find that cross-modal is again the best solution, as it matches its pre-training. Our method outperforms previous approaches, while being conceptually simpler and computationally cheaper. These performance gains are studied intrinsically according to the relevance of the retrieved documents and extrinsically according to the accuracy of the answer extracted by an external module. We discuss the differences between these metrics and their implications for KVQAE evaluation.

MOTS-CLÉS : questions visuelles, multimodalité, recherche cross-modale, entités nommées.

KEYWORDS: visual question answering, multimodality, cross-modal retrieval, named entities.

Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA), pages 74–92.

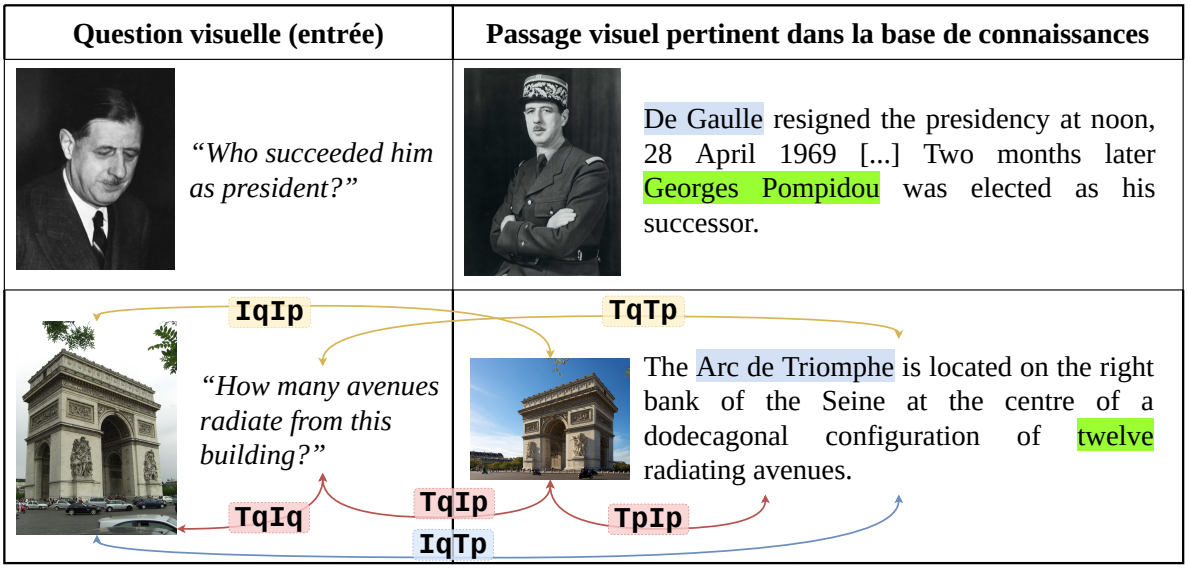


FIGURE 1 – Deux exemples de questions visuelles du jeu de données ViQuAE accompagnées de passages visuels pertinents issus de sa base de connaissances ainsi que d'une illustration des différents types d'interactions mono- et cross-modales étudiés (montrée seulement pour la deuxième question). Les sigles des interactions sont composés des lettres T (Texte), I (Image), Q (question) et P (passage).

1 Introduction

Répondre à des questions visuelles à propos d'entités nommées (KVQAE [1]) est une tâche difficile qui demande d'analyser des données multimodales (Shah *et al.*, 2019; Lerner *et al.*, 2022, 2023). Les représentations multimodales d'entités nommées y sont centrales, ce qui lie cette tâche à la désambiguïsation multimodale d'entités nommées (Adjali *et al.*, 2020). La KVQAE, comme d'autres tâches multimodales, vise à fluidifier et rendre plus naturelle l'interaction entre l'utilisateur et la machine. Par exemple, en regardant un film, on peut se demander « *Où ai-je déjà vu cette actrice ?* » ou « *Est-ce qu'elle a déjà gagné un Oscar ?* » Un système de question-réponse multimodal nous éviterait alors la fastidieuse tâche de parcourir le générique du film et de chercher des informations à propos de ladite actrice. La Figure 1 montre deux exemples de questions visuelles ainsi que des passages visuels pertinents correspondants, tirés du jeu de données ViQuAE (Lerner *et al.*, 2022) et de sa Base de Connaissances (BC) multimodale. Une question visuelle est constituée plus précisément d'une question textuelle, accompagnée d'une image qui lui est liée, et symétriquement, un passage visuel est la réunion d'un passage textuel, issu d'un document lié à une entité, et d'une image associée. Par ailleurs, à la différence de sa définition la plus courante en représentation des connaissances, le terme de *Base de Connaissances* renvoie ici à un contenu non structuré, formé de passages visuels.

Contrairement aux questions visuelles classiques (Antol *et al.*, 2015), qui visent le contenu de l'image (par exemple « *De quelle couleur est la voiture ?* »), les questions en KVQAE visent des entités nommées et nécessitent donc de rechercher des informations dans une BC. D'autres travaux se situent à mi-chemin mais sont limités à des catégories d'objet « gros grain », par exemple « personne » et « monument » au lieu de « Charles de Gaulle » et « Arc de Triomphe » (Marino *et al.*, 2019). Dans cette étude, nous définissons deux types d'interactions mono-modales, textuelle (TQTP) et visuelle (IQIP) entre question et passage, ainsi que trois cross-modales : au sein de la question visuelle (TQIQ), du

1. *Knowledge-based Visual Question Answering about named Entities*.

passage visuel (TPIP) ou entre les deux [2] (IQTP), comme illustré par la Figure 1. Nous nous focalisons principalement sur cette dernière interaction. La KVQAE ayant été introduite très récemment, elle reste largement à explorer. Shah *et al.* (2019) et Lerner *et al.* (2022) l'ont abordée en s'appuyant sur des représentations spécialisées pour les images, en l'occurrence en lien avec les visages, tandis que Lerner *et al.* (2023) ont proposé une méthode de pré-entraînement pour la Recherche d'Information (RI) multimodale. Leur méthode combine implicitement recherche mono- et cross-modale mais demande un pré-entraînement coûteux et n'exploite pas les modèles pré-entraînés existant pour la recherche cross-modale, tels que CLIP (Radford *et al.*, 2021).

CLIP est fondé sur un double encodeur, un pour chaque modalité, entraîné à partir d'images et de leurs légendes de façon cross-modale. Cette architecture permet d'être très efficient à la fois pour la recherche cross-modale mais aussi pour l'entraînement dans un cadre d'apprentissage contrastif, c'est-à-dire en comparant les représentations d'exemples selon leur similarité sémantique. CLIP s'est imposé comme un modèle fondateur (Bommasani *et al.*, 2021) de par ses multiples applications en vision par ordinateur, traitement automatique des langues et recherche cross-modale (Radford *et al.*, 2021; Ramesh *et al.*, 2021; Mokady *et al.*, 2021; Wolfe & Caliskan, 2022). Dans ce travail, nous montrons comment aborder la KVQAE grâce à CLIP, à la fois pour la recherche mono-modale mais aussi cross-modale, les deux pouvant être combinées aisément. Lerner *et al.* (2022) ont utilisé l'encodeur visuel de CLIP (CLIP_V) pour la recherche visuelle mais pas son encodeur textuel (CLIP_T). Lerner *et al.* (2023) ont combiné CLIP_V et BERT (Devlin *et al.*, 2019) pour la RI multimodale mais suggèrent que les bénéfices de leur modèle viennent des interactions cross-modales entre l'image de la question et le texte du passage visuel (IQTP, en bleu dans la Figure 1). Or, celles-ci sont possibles directement avec CLIP_T. De plus, la méthode de pré-entraînement de Lerner *et al.* (2023) est coûteuse. Nous montrons ici comment ajuster CLIP avec le peu de données de ViQuAE (Lerner *et al.*, 2022), sans pré-entraînement supplémentaire.

Nos résultats suggèrent que la recherche cross-modale est supérieure à la mono-modale, ce qui est intéressant car les modèles cross-modaux tels que CLIP peuvent être entraînés de façon faiblement supervisée. De plus, nous montrons que les deux sortes de recherche peuvent être combinées très facilement, sans entraînement supplémentaire, et que cette recherche hybride apporte des gains de performance conséquents. Enfin, nos expérimentations suggèrent que l'ajustement (*fine-tuning*) de CLIP est plus efficace en condition cross-modale, comme pour son pré-entraînement. Les gains de performance sont étudiés intrinsèquement selon la pertinence des résultats de la RI et extrinsèquement selon l'exactitude de la réponse extraite par un module externe. Nous discutons des différences entre ces métriques.

2 Travaux connexes

Notre travail se situe à l'intersection de plusieurs domaines de la RI : textuelle, visuelle et cross-modale, appliquée à des questions ou plus généralement à des requêtes. L'apprentissage de représentations denses pour la RI est transverse à ces trois domaines. Avec une représentation adéquate, la RI se ramène alors à un problème de recherche des plus proches voisins.

2. Nous ne nous intéressons pas à l'interaction TQIP entre l'image de la question et le texte du passage comme expliqué à la Section 3.

Recherche cross-modale Couairon *et al.* (2022) explorent les possibilités d'analogie multimodale avec les représentations de CLIP. Par exemple, la reine est au roi ce qu'une photo de femme est à une photo d'homme. Ils montrent que les représentations de CLIP ne sont pas à l'origine adaptées pour cet usage mais peuvent être ajustées facilement avec une simple projection linéaire en gardant le même objectif d'entraînement. Puisque ces analogies prennent la forme d'opérations arithmétiques sur des vecteurs, elles combinent recherche mono- et cross-modale de manière assez semblable à notre travail. Sun *et al.* (2022) emploient CLIP pour la recherche cross-modale dans le cadre d'une tâche connexe à la KVQAE : la désambiguïsation visuelle d'entités nommées. Ils montrent que CLIP surpasse un modèle de reconnaissance faciale, même en *zero-shot*. Nos résultats suggèrent le contraire mais Sun *et al.* (2022) utilisent un modèle de reconnaissance faciale différent du nôtre et ne précisent pas la version ni la taille du modèle CLIP avec lequel ils expérimentent. Leur travail se focalise sur le jeu de données qu'ils proposent et n'emploie ainsi CLIP que de manière cross-modale, avec un ajustement laissant les encodeurs fixes et donc, en ajoutant un perceptron multi-couches. Il reprend en cela le modèle CLIP-Adapter de Gao *et al.* (2021), qui supposent que l'ajustement de l'ensemble des paramètres de CLIP conduirait inévitablement au sur-apprentissage. Toutefois, Gao *et al.* (2021) ne vérifient pas cette hypothèse et se comparent principalement aux approches de *prompting*. Au contraire de Sun *et al.* (2022), nous utilisons CLIP à la fois pour la recherche mono- et cross-modale et ajustons l'ensemble de ses paramètres sans en introduire de nouveaux. Wang *et al.* (2022) utilisent quant à eux un objectif d'apprentissage contrastif à la fois mono- et cross-modal ayant pour fin la recherche cross-modale. Contrairement à notre cadre où l'on vise à rapprocher une image du nom de l'entité représentée et de son image de référence, les auteurs regroupent les représentations mono-modales par catégorie « gros grain » (par exemple les 20 catégories du jeu de données Pascal ; Rashtchian *et al.*, 2010).

Questions cross-modales Liu *et al.* (2023) travaillent sur le jeu de questions cross-modales WebQA (Chang *et al.*, 2022). Bien que WebQA ait à l'origine été proposé comme une évaluation de *reading comprehension*, Liu *et al.* (2023) traitent ses questions cross-modales sorties de leur contexte et travaillent donc sur la partie RI, visant justement à retrouver le contexte pertinent. On pourrait ainsi qualifier WebQA de « questions visuelles sans images » car les questions sont purement textuelles mais les réponses se trouvent dans une image. Par exemple, on peut répondre « *helmet* » à la question « *What is the sculpted bust at the Baroque library, Prague wearing on its head ?* » à condition de trouver une image pertinente. Liu *et al.* (2023) utilisent CLIP pour chercher des images à partir de la question. Ils exploitent également la légende des images et combinent ainsi les informations de manière similaire à la nôtre (cf. Section 3), sauf que leur requête est textuelle et non pas visuelle.

Questions visuelles de sens commun Gui *et al.* (2022) intègrent CLIP dans un encodeur-décodeur T5 (Raffel *et al.*, 2020) entraîné à générer la réponse à la manière de Lewis *et al.* (2020). CLIP, qui reste fixé, sert alors à la recherche cross-modale entre l'image de la question et le nom d'une entité accompagné de sa description dans un sous-graphe de Wikidata. Les auteurs expérimentent avec le jeu de données OK-VQA (Marino *et al.*, 2019), qui se focalise sur des questions de sens commun (*commonsense*) et vise ainsi des catégories d'objets « gros grain » plutôt que des entités nommées.

Questions visuelles à propos d'entités nommées (KVQAE) Garcia-Olano *et al.* (2022) et Heo *et al.* (2022) travaillent sur la KVQAE avec le jeu de données de Shah *et al.* (2019). Cependant, il est difficile de comparer leurs approches à la nôtre car leurs systèmes prennent en entrée la légende de

l'image, ce qui rend l'image elle-même redondante. Shah *et al.* (2019) ont proposé KVQA, le premier jeu de données pour la KVQAE. Ils traitent la multimodalité de la tâche par une fusion tardive au niveau de la décision : les entités nommées sont détectées et désambiguïsées à la fois dans la question et l'image par des modules indépendants avant d'être regroupées. Un sous-graphe de Wikidata est ensuite construit à partir de ces entités et traité par un *memory network* (Weston *et al.*, 2014). Notre travail est plus proche de Lerner *et al.* (2022), qui utilisent une BC fondée sur Wikipédia, faite donc d'images et de textes non-structurés (comme dans la Figure 1). Ils traitent la tâche en deux étapes, où l'extraction des réponses suit la RI. La RI est une combinaison de deux recherches mono-modales : textuelle avec DPR (Karpukhin *et al.*, 2020) et visuelle avec une combinaison de CLIP_V, ArcFace (Deng *et al.*, 2019) et un modèle ResNet entraîné sur ImageNet (He *et al.*, 2016; Deng *et al.*, 2009). Nous cherchons d'une part à simplifier ce système en supprimant la dépendance vis-à-vis d'ArcFace et ImageNet, deux modèles supervisés qui fournissent a priori des représentations moins génériques que CLIP, et d'autre part à exploiter pleinement CLIP en combinant recherche mono-modale et cross-modale. Après la RI, Lerner *et al.* (2022) extraient les réponses des passages de texte grâce à BERT multi-passage (Wang *et al.*, 2019). Pour leur part, Lerner *et al.* (2023) se sont, comme nous, focalisés sur la RI. Afin de modéliser les interactions cross-modales TQIQ et TPIP, ils représentent de façon jointe le texte et l'image, inspirés par les BERT multimodaux qui dominent les travaux sur les questions visuelles classiques ces dernières années (Khan *et al.*, 2022; Gan *et al.*, 2022). Néanmoins, ces architectures demandent un pré-entraînement coûteux et Lerner *et al.* (2023) suggèrent finalement que leur modèle exploite surtout l'interaction IQTP. Nos conclusions se rejoignent car notre modèle surpasse le leur — sans pré-entraînement supplémentaire — en modélisant explicitement IQTP via CLIP, comme décrit dans la section suivante.

3 Méthodes

Étant donné une question visuelle $(\mathbf{t_q}, \mathbf{i_q})$ et une BC consistant en une collection de passages visuels $(\mathbf{t_p}, \mathbf{i_p})$, nous cherchons à trouver des passages pertinents, c'est-à-dire permettant de répondre à la question. Nous nous concentrons ici sur les interactions cross-modales entre les questions et les passages. Nous laissons donc de côté les interactions cross-modales au sein des questions (TQIQ) et des passages (TPIP) (cf. Figure 1). Par ailleurs, nous ne considérons pas non plus la similarité TQIP entre la question et l'image du passage dans ce cadre car nous jugeons que la spécification de l'entité par le biais de la seule partie textuelle de la question est très peu discriminante du point de vue de l'entité référencée. Par conséquent, nous nous focalisons sur la recherche visuelle à partir de l'image $\mathbf{i_q}$. Pour ce faire, nous définissons la fonction de similarité suivante, qui combine similarités mono- et cross-modale (cf. Figure 1) :

$$s(\mathbf{i_q}, \mathbf{t_p}, \mathbf{i_p}) = \alpha_I s_I(\mathbf{i_q}, \mathbf{i_p}) + \alpha_C s_C(\mathbf{i_q}, \mathbf{t_p}) \tag{1}$$

où les paramètres $\alpha_{\{I,C\}}$ pondèrent chaque similarité. Cette décomposition nous permet d'exploiter directement des modèles pré-entraînés. Plus précisément, dans cette étude, nous nous focalisons sur l'ajustement de CLIP pour implémenter $s_I(\mathbf{i_q}, \mathbf{i_p})$ et $s_C(\mathbf{i_q}, \mathbf{t_p})$. L'objectif est donc de rapprocher l'image de la question de l'image de cette entité dans la BC (*optimisation mono-modale*) ou bien de son nom (*optimisation cross-modale*) ou les deux de façon jointe.

3.1 Objectif d'apprentissage et modèles

Plus formellement, l'objectif sous-tendant notre modèle de RI est de maximiser $s(\mathbf{i_q}, \mathbf{t_p}, \mathbf{i_p})$ si les deux images $\mathbf{i_q}$ et $\mathbf{i_p}^{(+)}$ représentent la même entité, nommée sous la forme textuelle $\mathbf{t_p}^{(+)}$, et de la minimiser sinon. Les données étant traitées par batch, ces entités négatives, pour lesquelles les représentations textuelles et visuelles sont notées respectivement $\mathbf{t_p}^{(j)}$ et $\mathbf{i_p}^{(j)}$, sont, dans une telle approche contrastive, constituées des autres entités du batch. Pour mettre en œuvre cette approche, nous optimisons de façon jointe $s_I(\mathbf{i_q}, \mathbf{i_p})$ et $s_C(\mathbf{i_q}, \mathbf{t_p})$ pour chaque image $\mathbf{i_q}$ du batch en minimisant l'objectif suivant, étant donné τ la température :

$$-\log \frac{\exp\left(s(\mathbf{i_q}, \mathbf{t_p}^{(+)}, \mathbf{i_p}^{(+)})e^{\tau}\right)}{\exp\left(s(\mathbf{i_q}, \mathbf{t_p}^{(+)}, \mathbf{i_p}^{(+)})e^{\tau}\right) + \sum_j \exp\left(s(\mathbf{i_q}, \mathbf{t_p}^{(j)}, \mathbf{i_p}^{(j)})e^{\tau}\right)} \tag{2}$$

Puisque nous implémentons $s_C(\mathbf{i_q}, \mathbf{t_p})$ avec CLIP :

$$s_C(\mathbf{i_q}, \mathbf{t_p}) = \cos\left(\mathrm{CLIP}_V(\mathbf{i_q}), \mathrm{CLIP}_T(\mathbf{t_p})\right) \tag{3}$$

Cet objectif correspond à celui utilisé pendant le pré-entraînement de CLIP si $\alpha_I = 0$ et $\alpha_C = 1$ (optimisation cross-modale seulement), sauf qu'il est asymétrique (la fonction softmax exprime les probabilités selon $\mathbf{i_q}$ et pas selon $\mathbf{t_p}$). Puisque $\mathbf{i_q}$, $\mathbf{t_p}$ et $\mathbf{i_p}$ sont encodés indépendamment, cet objectif permet d'exploiter toutes les autres images et textes du batch de manière très efficace (il suffit d'un produit matriciel pour calculer le dénominateur de l'équation 2). Nous implémentons $s_I(\mathbf{i_q}, \mathbf{i_p})$ de manière similaire : $\cos\left(\mathrm{CLIP}_V(\mathbf{i_q}), \mathrm{CLIP}_V(\mathbf{i_p})\right)$.

Les résultats de cette recherche visuelle peuvent être combinés avec la recherche textuelle $s_T(\mathbf{t_q}, \mathbf{t_p})$ en redéfinissant s de la façon suivante :

$$s(\mathbf{t_q}, \mathbf{i_q}, \mathbf{t_p}, \mathbf{i_p}) = \alpha_T s_T(\mathbf{t_q}, \mathbf{t_p}) + \alpha_I s_I(\mathbf{i_q}, \mathbf{i_p}) + \alpha_C s_C(\mathbf{i_q}, \mathbf{t_p}) \tag{4}$$

Nous discutons des difficultés à optimiser ces trois similarités de façon jointe dans la Section 4. De ce fait, $s_T(\mathbf{t_q}, \mathbf{t_p})$ est implémenté par un modèle entraîné séparément et les poids $\alpha_{\{T,I,C\}}$ sont déterminés par dichotomie sur le jeu de validation pour maximiser le rang réciproque moyen en contraignant leur somme à 1.

3.2 Baselines

Nous comparons notre approche au modèle de Lerner *et al.* (2022), qui combine DPR, CLIP_V, ArcFace et un modèle ResNet entraîné sur ImageNet. DPR est fondé sur deux encodeurs BERT (Devlin *et al.*, 2019) [3] : un pour la question et un pour le passage. Dans notre cas, il implémente $s_T(\mathbf{t_q}, \mathbf{t_p}) = \mathrm{DPR}(\mathbf{t_q}, \mathbf{t_p}) = \mathrm{BERT}_q(\mathbf{t_q})_{[\mathrm{CLS}]} \cdot \mathrm{BERT}_p(\mathbf{t_p})_{[\mathrm{CLS}]}$. Il est d'abord pré-entraîné sur TriviaQA (Joshi *et al.*, 2017) avant d'être ajusté sur ViQuAE. Les autres modèles sont disponibles publiquement et ne sont pas ajustés [4]. Les résultats des quatre modèles sont combinés de la même façon que dans l'équation 4, où DPR implémente $s_T(\mathbf{t_q}, \mathbf{t_p})$; CLIP_V, ArcFace, et ImageNet composent

3. `bert-base-uncased` disponible dans la bibliothèque Transformers.
4. ArcFace est disponible à https://github.com/deepinsight/insightface et ImageNet dans `torchvision`. Les deux utilisent une architecture ResNet-50.

Modèle	Mono-modal		Cross-modal		
	TQTP	IQIP	TQIQ	TPIP	IQTP
DPR	✓				
DPR + CLIP mono-modal *zero-shot*	✓	✓			
DPR et reconnaissance faciale (Lerner *et al.*, 2022)	✓	✓			
ECA (Lerner *et al.*, 2023)	✓	✓	✓	✓	✓
ILF (Lerner *et al.*, 2023)	✓	✓			✓
DPR + CLIP mono- et cross-modal ajusté	✓	✓			✓

TABLE 1 – Récapitulatif des différentes interactions mono- et cross-modales utilisées par les modèles étudiés.

$s_I(\mathbf{i_q}, \mathbf{i_p})$ et il n'y a pas de similarité cross-modale ; donc $s_C(\mathbf{i_q}, \mathbf{t_p}) = 0$. Plus précisément, ArcFace est utilisé de manière alternative à CLIP_V et ImageNet : seulement lorsqu'un visage est détecté. La recherche est alors effectuée seulement sur les entités nommées de type personne dans la BC, en supposant que les visages sont pertinents seulement pour les personnes. Formellement, en notant ArcFace A, CLIP_V V, ImageNet R, $F \in \{0, 1\}$ la détection d'un visage dans $\mathbf{i_q}$ et $\mathbf{i_p}$ et $H \in \{0, 1\}$ si $\mathbf{i_p}$ correspond à une personne [5] :

$$s_I(\mathbf{i_q}, \mathbf{i_p}) = FH\alpha_A s_A(\mathbf{i_q}, \mathbf{i_p}) + (1 - F)(1 - H)\left(\alpha_V s_V(\mathbf{i_q}, \mathbf{i_p}) + \alpha_R s_R(\mathbf{i_q}, \mathbf{i_p})\right) \tag{5}$$

Pour rendre les scores de ces différents modèles comparables, ils sont centrés-réduits. De plus, quand un document n'est pas retrouvé par un système donné (mais par les autres, puisqu'on considère toujours le top-K d'un système), on lui assigne le score minimal des autres résultats de ce système, selon la technique du « minimum par défaut » de Ma *et al.* (2021).

Nous nous comparons également aux modèles ECA et ILF de Lerner *et al.* (2023). ECA (*Early Cross-Attention*) fusionne les modalités de manière précoce à l'aide d'un mécanisme d'attention, comme son nom l'indique. La similarité est donc calculée suivant $s(\mathbf{t_q}, \mathbf{i_q}, \mathbf{t_p}, \mathbf{i_p}) = \text{ECA}(\mathbf{t_q}, \mathbf{i_q}) \cdot \text{ECA}(\mathbf{t_p}, \mathbf{i_p})$ et combine ainsi toutes les interactions multimodales présentées à la Figure 1. ILF (*Intermediate Linear Fusion*) fusionne les modalités avec une simple projection linéaire et n'a donc, comme notre méthode, ni interaction TQIQ ni interaction TPIP puisque la similarité s'y réduit à :

$$s(\mathbf{t_q}, \mathbf{i_q}, \mathbf{t_p}, \mathbf{i_p}) = s_T(\mathbf{t_q}, \mathbf{t_p}) + s_{C'}(\mathbf{t_q}, \mathbf{i_p}) + s_I(\mathbf{i_q}, \mathbf{i_p}) + s_C(\mathbf{i_q}, \mathbf{t_p}) \tag{6}$$

Les différentes interactions mono- et cross-modales utilisées par les modèles étudiés sont résumées dans le Tableau 1.

Il est à noter que Lerner *et al.* (2022) et Lerner *et al.* (2023) emploient CLIP_V avec l'architecture ResNet tandis que nous utilisons ViT (Dosovitskiy *et al.*, 2021) dans la plupart de nos expériences (mais comparons les deux dans la Section 5 sans trouver de différence significative) [6].

5. On connaît seulement le type d'entité des images $\mathbf{i_p}$ de la BC, *pas* celles des questions $\mathbf{i_q}$.
6. Plus précisément, il s'agit de `RN50` et `ViT-B/32` disponibles à `https://github.com/openai/CLIP`

4 Implémentation

4.1 Données

Nous utilisons la BC proposée par Lerner *et al.* (2022), qui consiste en 1,5 millions d'articles Wikipédia et images des entités Wikidata correspondantes. Les articles sont divisés en 12 millions de passages de 100 mots. Par conséquent, tous les passages d'un même article partagent la même image. Deux exemples de passages visuels sont montrés à la Figure 1. Par la suite, nous évaluons les méthodes à deux niveaux de RI : article et passage.

Notre étude se focalise sur ViQuAE, un des deux seuls jeux de données pour le KVQAE. Nous n'expérimentons pas avec l'autre, KVQA (Shah *et al.*, 2019), pour les mêmes raisons que Lerner *et al.* (2023) : KVQA ayant été généré automatiquement à partir de Wikidata, rien ne garantit que les réponses se trouvent dans la BC. De plus, il comprend 29 % de questions booléennes (réponse oui/non) pour lesquelles la pertinence du passage ne peut pas être évaluée automatiquement sur la base de la présence de la réponse à la question.

ViQuAE contient 3 700 questions visuelles à propos de 2 400 entités différentes, réparties aléatoirement en ensembles de taille égale pour l'entraînement, la validation et le test, sans recouvrement entre les images. Par conséquent, le recouvrement entre les entités du jeu d'entraînement et de test est très faible, seulement de 18 %. Nos modèles doivent donc apprendre à généraliser non seulement à de nouvelles images mais aussi à de nouvelles entités. Notons que tout le texte, des questions comme de la BC, est en anglais.

4.2 Problème de l'annotation de référence

Comme nous l'avons indiqué à la Section 3.1, l'optimisation de la similarité textuelle entre question et passage $s_T(\mathbf{t_q}, \mathbf{t_p})$, implémentée par DPR (cf. Section 3.2), avec les deux autres similarités ne s'est pas accompagnée dans nos expériences d'améliorations significatives au niveau des résultats. Nous interprétons ce constat comme une conséquence des incohérences induites par l'annotation de référence des passages concernant la modalité visuelle, ce qui nuit à l'entraînement d'un modèle visuel. Plus précisément, tous les passages visuels du même article partageant la même image, celle-ci peut être considérée comme pertinente ou non pertinente selon le texte qui lui est associé. De plus, la même image (ou deux images de la même entité) peut illustrer deux articles différents, donc encore une fois avoir une pertinence variable selon le texte associé. À l'inverse, on peut trouver un passage pertinent dans un autre article que celui de l'entité-sujet, donc illustré par une image très différente, mais qui sera alors considérée comme pertinente. À cause de ces difficultés, nous avons opté pour une autre annotation, indépendante du passage. Il est néanmoins intéressant de constater que Lerner *et al.* (2023) ont réussi à entraîner leurs modèles, ECA et ILF (cf. Section 3.2), avec l'annotation au niveau du passage. Ce succès pourrait être expliqué par la représentation jointe d'ECA (qui modélise TQIQ et TPIP) ou par l'expressivité d'ILF. Une autre explication, pas forcément incompatible, serait liée à l'interaction IQTP, car ECA et ILF considèrent le passage entier tandis que CLIP n'est appliqué qu'au titre de l'article.

À la place de cette annotation au niveau du passage, nous utilisons l'annotation au niveau de l'entité fournie par Lerner *et al.* (2022) car chaque question visuelle porte sur une seule et unique entité. Pour ce faire, nous retirons 25 questions visuelles du jeu d'entraînement de ViQuAE pour le réduire à

1 165 car les entités correspondantes sont absentes de la BC [7].

4.3 Hyperparamètres

Pour profiter au mieux des entités associées aux autres images du batch $\mathbf{t_p}^{(j)}$ et $\mathbf{i_p}^{(j)}$, nous utilisons un batch de la plus grande taille possible, ici 1 165 triplets $(\mathbf{i_q}, \mathbf{t_p}^{(+)}, \mathbf{i_p}^{(+)})$, soit l'intégralité du jeu d'entraînement. Nous utilisons une seule GPU NVIDIA V100 avec 32 Go de mémoire vive. La grande taille de batch est en partie permise par le *gradient checkpointing*.

Puisque le jeu d'entraînement est petit, l'entraînement est très peu coûteux : notre meilleur modèle converge [8] au bout de 11 époques/itérations, en moins de 15 minutes, ce qui est négligeable par rapport au pré-entraînement de 8 000 itérations en trois jours de Lerner *et al.* (2023) avec le même matériel [9].

Nous utilisons un taux d'apprentissage très faible, de 2×10^{-6}, croissant linéairement pendant 4 époques puis décroissant pendant 46 époques, si l'entraînement n'est pas interrompu avant. L'optimisation est faite avec AdamW (Loshchilov & Hutter, 2019), avec $\lambda = 0{,}1$. Pour l'optimisation jointe, nous initialisons $\alpha_I = \alpha_C = 0{,}5$ et leur assignons un taux d'apprentissage de $0{,}02$, beaucoup plus grand que le reste du modèle. À l'instar de Radford *et al.* (2021), la température τ reste entraînable mais, étant donné le faible taux d'apprentissage, elle reste proche de sa valeur initiale, soit $4{,}6$ [10]. Ces hyperparamètres ont été déterminés manuellement sur le jeu de validation.

L'entraînement est interrompu et le meilleur modèle sélectionné selon le meilleur rang réciproque moyen au sein du batch sur le jeu de validation, c'est-à-dire en réordonnant les images ou textes du batch selon le score de similarité s, pour éviter de calculer les représentations de toute la BC à chaque époque.

Notre implémentation est fondée sur Lightning [11], PyTorch (Paszke *et al.*, 2019) et Transformers (Wolf *et al.*, 2020) pour l'entraînement des modèles, et Datasets (Lhoest *et al.*, 2021), Faiss (Johnson *et al.*, 2019) et Ranx (Bassani, 2022) pour la RI. Notre code est disponible librement à `https://github.com/PaulLerner/ViQuAE`.

5 Résultats

Nous évaluons la RI à deux niveaux :

— article (qui contient plusieurs passages ; cf. Section 4.1) ;

— passage visuel, afin de pouvoir nous comparer aux autres méthodes (Lerner *et al.*, 2022, 2023).

Dans les deux cas, un document (passage ou article entier) est jugé pertinent s'il contient la réponse après prétraitement standard (insensibilité à la casse, aux déterminants et à la ponctuation). Les métriques utilisées sont la précision à K (P@K) et le rang réciproque moyen (MRR) ainsi que Hits@K

7. Parce qu'elles n'ont pas d'images libres de droit.
8. C'est-à-dire commence à sur-apprendre.
9. Lerner *et al.* (2023) rapportent un bilan carbone de 1,7 kgCO2e pour trois jours de consommation électrique des GPUs.
10. Nous avons gardé la formulation de Radford *et al.* (2021) mais la température est habituellement exprimée sous la forme $\frac{1}{\tau'}$ et non pas e^{τ}, ce qui équivaudrait à $\tau' = \frac{1}{100}$ ici.
11. `https://www.pytorchlightning.ai/`

Recherche	Optimisation	MRR	P@1	P@20	Hits@20
Mono-modale (IQIP)	Non (*zero-shot*)	29,4	21,8	9,1	53,4
	Mono-modale	30,0	21,8	9,2	55,7
	Cross-modale	29,8	21,4	**9,5**	54,7
	Jointe	**30,4**	**22,0**	**9,5**	**55,8**
Cross-modale (IQTP)	Non (*zero-shot*)	32,7	23,1	10,9	60,6
	Mono-modale	31,6	21,9	10,9	59,6
	Cross-modale	**37,1**	**26,9**	**11,9**	**67,8**
	Jointe	30,8	21,3	10,4	59,5
Fusion	Non (*zero-shot*)	39,6	30,6	11,8	63,9
	Mono-modale	40,1	31,8	11,6	63,6
	Cross-modale	***44,1***	***34,9***	***12,7***	***69,9***
	Jointe	41,0	32,6	11,6	64,9
	Disjointe	43,7	34,5	***12,7***	***69,9***

TABLE 2 – Validation des différentes méthodes d'ajustement de CLIP (ainsi que la version *zero-shot* pour référence) pour la recherche visuelle (à partir de l'image de la question $\mathbf{i_q}$). L'évaluation est faite ici au niveau de l'article sur le sous-ensemble de validation de ViQuAE. Pour chaque recherche (mono- ou cross-modale), les meilleurs résultats sont marqués en gras. Les meilleurs résultats au total sont obtenus par la fusion des deux et sont marqués en gras italique. Fusion de l'optimisation disjointe : recherche mono-modale optimisée de manière mono-modale et idem pour cross-modale.

(équivalent au rappel en considérant qu'il n'y a qu'un seul document pertinent par question visuelle). P@1 et Hits@1 sont équivalents.

Une fois la RI effectuée au niveau du passage visuel, les réponses sont extraites à l'aide du modèle BERT multi-passage entraîné par Lerner *et al.* (2022). Deux métriques sont utilisées pour évaluer les réponses : l'appariement exact et le score F1 (au niveau des sacs de mots) entre la réponse extraite et la vérité terrain [12].

5.1 Recherche d'information au niveau de l'article

Nous explorons dans un premier temps trois modes d'optimisation et trois manières d'utiliser CLIP au travers d'expériences menées sur le jeu de validation. Ces trois modes peuvent être décrits à partir de l'équation 1 :

— recherche/optimisation mono-modale entre les deux images, soit $\alpha_I = 1, \alpha_C = 0$;
— recherche/optimisation cross-modale entre l'image et le nom de l'entité, soit $\alpha_I = 0, \alpha_C = 1$;
— fusion des deux recherches ou optimisation jointe, soit $\alpha_I > 0, \alpha_C > 0$.

À noter que le mode d'optimisation n'influence pas le mode de recherche, comme en témoigne le Tableau 2. Pour mémoire, CLIP est pré-entraîné de manière cross-modale uniquement (Radford *et al.*, 2021).

12. Ou plutôt les vérités terrains car les alias Wikipédia d'une entité constituent une réponse valide.

Recherche mono- ou cross-modale ? Avant de comparer les différentes méthodes d'optimisation, nous pouvons d'ores et déjà remarquer que la RI cross-modale l'emporte [13] systématiquement sur la RI mono-modale, notamment en *zero-shot* [14], ce qui peut paraître surprenant puisque les noms propres portent a priori peu de sémantique. On s'étonne donc que CLIP parvienne à généraliser [15] la représentation d'entités à partir de leurs seuls noms. Néanmoins, certains noms sont tout de même porteurs de sens. Par exemple, un nom peut indiquer le genre d'une personne et suggérer sa nationalité. De plus, nous travaillons ici avec les titres des articles Wikipédia, qui sont également susceptibles de contenir la nature de l'entité (par exemple la profession d'une personne ou le type de monument). Ces caractéristiques peuvent ainsi être mises en correspondance avec des attributs visuels. Enfin, nous attribuons principalement le succès de la RI cross-modale à son adéquation avec le pré-entraînement de CLIP : l'espace de représentation de CLIP est organisé pour rapprocher textes et images similaires, la proximité mono-modale des images n'en est qu'une conséquence indirecte.

Pourquoi choisir ? Nous montrons que les recherches mono- et cross-modales sont complémentaires : leurs résultats peuvent être simplement combinés au niveau du score (comme dans l'équation 1). Pour l'optimisation jointe, nous pourrions utiliser directement les poids α optimisés par descente de gradient avec le reste du modèle sur le jeu d'entraînement, mais cela détériore légèrement les résultats. Ainsi, en *zero-shot*, la fusion des deux recherches apporte une amélioration relative de 32 % en P@1 par rapport à la recherche cross-modale seulement (significatif avec $p \leq 0{,}01$). Il serait intéressant d'étudier si ces résultats se généralisent à d'autres tâches. Cette méthode pourrait par exemple bénéficier à la recherche visuelle par le contenu, dans un contexte de navigation sur le Web.

Quelle optimisation ? On peut voir que l'optimisation cross-modale améliore légèrement la recherche mono-modale mais pas l'inverse. Dans les trois cas, l'optimisation dans un mode améliore au moins la recherche dans le même mode. Il est intéressant de noter que l'optimisation jointe détériore la RI cross-modale mais améliore la fusion (toujours par rapport au *zero-shot*). Mais au total, l'optimisation cross-modale semble être la meilleure option, surpassant significativement l'optimisation mono-modale. Nous l'expliquons encore une fois largement par son adéquation avec le pré-entraînement de CLIP : nous manquons probablement de données pour réorganiser l'espace de représentations. Conséquemment, nous supposons que l'optimisation mono-modale pénalise l'optimisation jointe. On voit également que les différences entre les modes d'optimisation de la recherche mono-modale sont très faibles : il n'est pas bénéfique de combiner la recherche mono-modale optimisée de manière mono-modale et la recherche cross-modale optimisée de manière cross-modale (« optimisation disjointe » dans le Tableau 2). Par conséquent, dans la suite de l'article nous utilisons le modèle entraîné de manière cross-modale et présentons les résultats sur le jeu de test.

5.2 Recherche d'information au niveau du passage visuel

Les résultats sont présentés dans le Tableau 3. Nous utilisons comme *baseline* DPR (recherche avec la question seulement) ainsi que sa fusion avec la recherche mono-modale de CLIP *zero-shot* (résultats rapportés par Lerner *et al.*, 2023). Puisque ces résultats, ainsi que ceux des autres modèles de Lerner

13. Significativement selon le test de randomisation de Fisher avec $p \leq 0{,}01$ (Fisher, 1937; Smucker *et al.*, 2007).
14. C'est-à-dire sans ajustement sur ViQuAE.
15. À moins que son jeu de pré-entraînement ne contienne suffisamment d'entités de ViQuAE pour que ce ne soit pas nécessaire. Nous développons cette discussion dans la Section 7.

Modèle	MRR	P@1	P@20	Hits@20
DPR	32,8	22,8	16,4	61,2
DPR + CLIP mono-modal *zero-shot*	34,5	24,8	15,8	61,8
DPR + CLIP* mono-modal *zero-shot*	34,7	24,3	16,0	62,8
DPR et reconnaissance faciale (Lerner *et al.*, 2022)	**37,9**	27,8	17,5	65,7
ECA (Lerner *et al.*, 2023)	37,8	26,7	**19,5**	**67,6**
ILF (Lerner *et al.*, 2023)	37,3	26,8	19,1	66,9
DPR + CLIP* mono- et cross-modal ajusté	37,6	**28,6**	16,3	63,6

TABLE 3 – Résultats de la RI évaluée au niveau du passage visuel sur le jeu de test de ViQuAE. *CLIP fondé sur l'architecture ViT au lieu de ResNet.

Recherche d'Information	Appariement exact	F1
DPR	16,9 ± 0,4	20,1 ± 0,5
DPR + CLIP mono-modal *zero-shot*	19,0 ± 0,4	22,3 ± 0,4
DPR + CLIP* mono-modal *zero-shot*	19,7 ± 0,9	23,3 ± 0,8
DPR et reconnaissance faciale (Lerner *et al.*, 2022)	22,1 ± 0,5	25,4 ± 0,4
ECA (Lerner *et al.*, 2023)	20,6 ± 0,3	24,4 ± 0,2
ILF (Lerner *et al.*, 2023)	21,3 ± 0,6	25,4 ± 0,3
DPR + CLIP* mono- et cross-modal ajusté	**24,7** ± 0,5	**28,7** ± 0,4

TABLE 4 – Résultats de l'extraction des réponses sur le jeu de test de ViQuAE. Moyennes sur 5 entraînements du modèle d'extraction avec des graines aléatoires différentes. Ce modèle prend en entrée le top-24 des différents systèmes de RI. *CLIP fondé sur l'architecture ViT au lieu de ResNet.

et al. (2022) et Lerner *et al.* (2023), sont fondés sur l'architecture ResNet pour CLIP, nous ajoutons également les résultats obtenus avec l'architecture ViT, utilisée dans le reste de nos expériences. Les deux architectures fournissent des résultats similaires.

Notre méthode améliore la précision@1 de 3 % relativement au modèle de Lerner *et al.* (2022), sans utiliser ArcFace, ni ImageNet, ni l'heuristique de la division de la BC entre personnes et non-personnes, et de 7 % relativement aux modèles de Lerner *et al.* (2023), sans pré-entraînement supplémentaire. Les différences de MRR avec ces modèles sont très faibles, mais ECA et ILF surpassent notre méthode en P@20 et Hits@20, ce qui suggérerait un avantage de la représentation jointe d'ECA et de l'expressivité d'ILF. Nous discutons davantage de ces métriques dans la Section 6.

On peut voir que les améliorations par rapport à la baseline *DPR + CLIP mono-modal zero-shot* sont assez modestes, beaucoup plus faibles que dans la section précédente où nous étudions les résultats au niveau de l'article et avant la fusion avec DPR. Nous verrons dans la section suivante que l'impact sur l'extraction des réponses est, lui, plus important, et démontre la supériorité de notre approche.

5.3 Extraction des réponses

Nous suivons le même protocole que Lerner *et al.* (2023) pour extraire les réponses, c'est-à-dire que nous utilisons le modèle fourni par Lerner *et al.* (2022), pré-entraîné sur TriviaQA puis ajusté sur

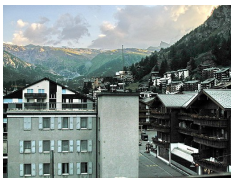
Question visuelle (entrée)	DPR + CLIP mono- et cross-modal ajusté	ECA
 *"In which state of the USA would you find this National Park?*	 Yosemite National Park is located in the central Sierra Nevada of California [...]	 Udall oversaw the addition of four national parks [...] including Canyonlands National Park in Utah, North Cascades National Park in Washington, Redwood National Park in California, the Great Swamp National Wildlife Refuge in New Jersey [...]
 *"This municipality is a ski resort in which European country?"*	 Zermatt and Saas-Fee have both summer ski areas. [...] the majority of ski resorts in Switzerland tend to open in December and run through to April.	 Major ski resorts are located mostly in the various European countries (e.g. Andorra, Austria, Bulgaria, Bosnia-Herzegovina, Croatia, Czech Republic, [...], Poland, Romania, Serbia, Sweden, Slovakia, Slovenia, Spain, Switzerland, Turkey) [...]

FIGURE 2 – Exemples qualitatifs où BERT multi-passage parvient à extraire la réponse du passage pertinent fourni par notre méthode de RI tandis qu'il est distrait par le passage fourni par ECA (Lerner *et al.*, 2023), qui contient beaucoup de réponses plausibles mais n'est pas vraiment pertinent (tout en étant considéré comme tel car il contient la réponse).

ViQuAE, qui prend 24 passages en entrée, lesquels varient selon les systèmes de RI.

Les résultats sont présentés dans le Tableau 4. On voit que la recherche cross-modale et l'ajustement de CLIP apportent 23 % à 25 % d'amélioration relativement à la baseline *DPR + CLIP mono-modal zero-shot* selon les métriques, ce qui est plus cohérent avec les résultats de la RI au niveau de l'article (cf. Section 5.1) où nous avions alors de 31 % à 60 % d'amélioration relative selon les métriques (avant la fusion avec DPR) [16]. Ainsi, notre méthode apporte des améliorations appréciables, de 12 % à 20 % selon les métriques et modèles : par rapport au modèle de Lerner *et al.* (2022), sans utiliser ArcFace, ni ImageNet, ni l'heuristique de la division de la BC entre personnes et non-personnes, et par rapport aux modèles de Lerner *et al.* (2023), sans pré-entraînement supplémentaire.

Nous discutons davantage de ces résultats et des différences entre les métriques dans la section suivante.

6 Discussion

La section précédente rapporte des différences importantes entre les métriques de RI au niveau du passage visuel et de l'extraction des réponses. Notre système peut être replacé dans le cadre de l'apprentissage augmenté par RI défini par Zamani *et al.* (2022). Les métriques de RI constituent alors une évaluation intrinsèque des modèles de RI tandis que l'extraction de réponse est une évaluation extrinsèque. Il est intéressant de noter que les métriques de RI que nous utilisons ont été conçues pour des utilisateurs humains et que les modèles d'apprentissage exploitent les résultats de la RI de

16. La Section 5.1 présente les résultats sur le jeu de validation mais ils sont similaires sur le jeu de test.

manière assez différente. Par exemple, le modèle BERT multi-passage ne tient pas compte du rang du passage visuel, les top-K passages étant traités en parallèle.

De plus, les métriques d'extraction de réponse sont moins sensibles au biais textuel inhérent à la KVQAE. Pour reprendre le deuxième exemple de la Figure 1, « *Combien y a-t-il d'avenues autour de ce bâtiment ?* », on peut énumérer les chiffres de 1 à 20, sans regarder l'image, et obtenir ainsi un $\text{Hits@20} = 1$. Néanmoins, un modèle d'extraction de réponse tel que BERT multi-passage aura seulement une chance sur 20 environ d'extraire la bonne réponse car elles sont toutes plus ou moins plausibles, comme discuté dans Lerner *et al.* (2022). Au contraire, avec une RI purement visuelle, donc exempte de biais textuel, on récupère les passages de l'article Wikipédia de l'Arc de Triomphe ou d'autres monuments ressemblants. Les métriques de RI sont alors mauvaises car la plupart des passages ne sont pas pertinents mais il suffit d'un seul passage pertinent dans le top-24 pour que BERT multi-passage puisse extraire la réponse sans ambiguïté car seuls les passages pertinents fournissent alors des réponses plausibles.

On peut observer ce phénomène quantitativement : entre la recherche purement textuelle (DPR) et la *baseline* multimodale (DPR + CLIP mono-modal *zero-shot*), il n'y a presque pas de différence pour les métriques de RI au niveau du passage (cf. Tableau 3), voire une détérioration de la précision@20, alors que les métriques d'extraction de réponse (cf. Tableau 4) montrent 16 % à 17 % d'amélioration relative pour la RI multimodale *baseline*. De la même manière, les modèles de fusion précoce de Lerner *et al.* (2023) sont plus à même d'exploiter les biais textuels que CLIP, qui cherche seulement à partir de l'image. Deux exemples sont montrés à la Figure 2.

Bien que cette évaluation extrinsèque corrige certains biais de l'évaluation intrinsèque, puisqu'elle repose sur un modèle externe, elle ajoute aussi plusieurs facteurs, notamment :

— l'architecture du modèle d'extraction (ici BERT multi-passage)
— son entraînement, notamment les données et le système de RI utilisés (ici *DPR et reconnaissance faciale*)
— le top-K (ici 24)

Ces questions sont interdépendantes. Par exemple, le top-K à l'inférence peut dépendre du nombre de passages utilisés pendant l'entraînement (24 aussi ici) ou de l'architecture : Lerner *et al.* ont tenté de fusionner le score d'extraction de réponse et de RI sans obtenir d'amélioration significative. L'étude de ces facteurs sort du cadre de cet article mais devrait être réalisée dans de futurs travaux.

7 Conclusion

Dans cet article, nous étudions la recherche cross-modale et sa combinaison avec la recherche mono-modale pour répondre à des questions visuelles à propos d'entités nommées (KVQAE), en nous focalisant sur le modèle CLIP. Nos résultats démontrent la supériorité de la recherche cross-modale, mais aussi la complémentarité des deux, qui peuvent être combinées facilement. Il serait intéressant d'étudier si ces résultats se généralisent à d'autres tâches. Cette méthode pourrait par exemple bénéficier à la recherche visuelle par le contenu dans un contexte de navigation Web. Bien que ce soit l'abondance de données cross-modales qui ait permis d'entraîner un modèle avec la capacité de CLIP, ce qui aurait été difficile avec une annotation mono-modale, cela limite nos résultats car il est difficile de contrôler une telle masse de données et donc d'estimer les capacités de généralisation de CLIP. Nous étudions également différentes manières d'ajuster CLIP et trouvons que l'optimisation cross-modale est la meilleure solution, encore une fois grâce à son adéquation avec son pré-entraînement.

Cette conclusion pourrait changer si nous disposions de suffisamment de données pour réorganiser l'espace de représentations.

Notre méthode surpasse la *baseline* (recherche mono-modale) mais aussi les méthodes de Lerner *et al.* (2022) et Lerner *et al.* (2023), tout en étant plus simple et moins coûteuse. Nos résultats questionnent toutefois les métriques utilisées pour évaluer les modèles de RI, notamment l'évaluation intrinsèque des passages, qui est sujette aux biais textuels. Nous préconisons donc de comparer prudemment des modèles fondés sur les mêmes données, comme dans la Section 5.1, ou bien d'évaluer extrinsèquement les résultats via un modèle d'extraction de réponse (Section 5.3). Toutefois, l'utilisation d'un modèle d'apprentissage pour évaluer les résultats est source de variabilité et pourrait donc changer les conclusions d'une étude. Nous avons notamment identifié trois facteurs importants dans la Section 6 qui devraient être étudiés dans de futurs travaux. Il serait également intéressant d'étudier l'optimisation jointe de la RI et de l'extraction/génération de réponse. De récents travaux ont montré sa faisabilité pour des tâches connexes à la KVQAE (Chen *et al.*, 2022; Hu *et al.*, 2022).

Remerciements

Les auteurs remercient chaleureusement les relecteurs anonymes pour leur retour constructif ainsi qu'Antoine Chaffin pour les discussions à propos de CLIP et de la recherche cross-modale. Ce travail a été financé par le projet ANR-19-CE23-0028 MEERQAT. Il a en outre bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2022-AD011012846R1 attribuée par GENCI.

Références

ADJALI O., BESANÇON R., FERRET O., LE BORGNE H. & GRAU B. (2020). Multimodal Entity Linking for Tweets. In J. M. JOSE, E. YILMAZ, J. MAGALHÃES, P. CASTELLS, N. FERRO, M. J. SILVA & F. MARTINS, Éds., *Advances in Information Retrieval*, Lecture Notes in Computer Science, p. 463–478, Cham : Springer International Publishing. DOI : 10.1007/978-3-030-45439-5_31.

ANTOL S., AGRAWAL A., LU J., MITCHELL M., BATRA D., ZITNICK C. L. & PARIKH D. (2015). VQA : Visual Question Answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, p. 2425–2433, Santiago, Chile : IEEE. DOI : 10.1109/ICCV.2015.279.

BASSANI E. (2022). ranx : A Blazing-Fast Python Library for Ranking Evaluation and Comparison. In M. HAGEN, S. VERBERNE, C. MACDONALD, C. SEIFERT, K. BALOG, K. NØRVÅG & V. SETTY, Éds., *Advances in Information Retrieval*, Lecture Notes in Computer Science, p. 259–264, Cham : Springer International Publishing. DOI : 10.1007/978-3-030-99739-7_30.

BOMMASANI R., HUDSON D. A., ADELI E., ALTMAN R., ARORA S., VON ARX S., BERNSTEIN M. S., BOHG J., BOSSELUT A., BRUNSKILL E., BRYNJOLFSSON E., BUCH S., CARD D., CASTELLON R., CHATTERJI N., CHEN A., CREEL K., DAVIS J. Q., DEMSZKY D., DONAHUE C., DOUMBOUYA M., DURMUS E., ERMON S., ETCHEMENDY J., ETHAYARAJH K., FEI-FEI L., FINN C., GALE T., GILLESPIE L., GOEL K., GOODMAN N., GROSSMAN S., GUHA N., HASHIMOTO T., HENDERSON P., HEWITT J., HO D. E., HONG J., HSU K., HUANG J., ICARD T., JAIN S., JURAFSKY D., KALLURI P., KARAMCHETI S., KEELING G., KHANI F., KHATTAB O., KOH P. W., KRASS M., KRISHNA R., KUDITIPUDI R., KUMAR A., LADHAK F., LEE M., LEE T.,

LESKOVEC J., LEVENT I., LI X. L., LI X., MA T., MALIK A., MANNING C. D., MIRCHANDANI S., MITCHELL E., MUNYIKWA Z., NAIR S., NARAYAN A., NARAYANAN D., NEWMAN B., NIE A., NIEBLES J. C., NILFOROSHAN H., NYARKO J., OGUT G., ORR L., PAPADIMITRIOU I., PARK J. S., PIECH C., PORTELANCE E., POTTS C., RAGHUNATHAN A., REICH R., REN H., RONG F., ROOHANI Y., RUIZ C., RYAN J., RÉ C., SADIGH D., SAGAWA S., SANTHANAM K., SHIH A., SRINIVASAN K., TAMKIN A., TAORI R., THOMAS A. W., TRAMÈR F., WANG R. E., WANG W., WU B., WU J., WU Y., XIE S. M., YASUNAGA M., YOU J., ZAHARIA M., ZHANG M., ZHANG T., ZHANG X., ZHANG Y., ZHENG L., ZHOU K. & LIANG P. (2021). On the Opportunities and Risks of Foundation Models. *arXiv :2108.07258 [cs]*. arXiv : 2108.07258.

CHANG Y., NARANG M., SUZUKI H., CAO G., GAO J. & BISK Y. (2022). Webqa : Multihop and multimodal qa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 16495–16504.

CHEN W., HU H., CHEN X., VERGA P. & COHEN W. (2022). MuRAG : Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 5558–5570, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics.

COUAIRON G., DOUZE M., CORD M. & SCHWENK H. (2022). Embedding arithmetic of multimodal queries for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, p. 4950–4958.

DENG J., DONG W., SOCHER R., LI L.-J., LI K. & FEI-FEI L. (2009). ImageNet : A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, p. 248–255. ISSN : 1063-6919, DOI : 10.1109/CVPR.2009.5206848.

DENG J., GUO J., XUE N. & ZAFEIRIOU S. (2019). Arcface : Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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.

DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISSENBORN D., ZHAI X., UNTERTHINER T., DEHGHANI M., MINDERER M., HEIGOLD G., GELLY S., USZKOREIT J. & HOULSBY N. (2021). An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale. In *Proceedings of ICLR 2021*.

FISHER R. A. (1937). The design of experiments. *The design of experiments.*, (2nd Ed). Publisher : Oliver & Boyd, Edinburgh & London.

GAN Z., LI L., LI C., WANG L., LIU Z. & GAO J. (2022). Vision-language pre-training : Basics, recent advances, and future trends. *Found. Trends. Comput. Graph. Vis.*, **14**(3–4), 163–352. DOI : 10.1561/0600000105.

GAO P., GENG S., ZHANG R., MA T., FANG R., ZHANG Y., LI H. & QIAO Y. (2021). CLIP-Adapter : Better Vision-Language Models with Feature Adapters. arXiv :2110.04544 [cs].

GARCIA-OLANO D., ONOE Y. & GHOSH J. (2022). Improving and diagnosing knowledge-based visual question answering via entity enhanced knowledge injection. In *Companion Proceedings of the Web Conference 2022*, WWW '22, p. 705–715, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3487553.3524648.

GUI L., WANG B., HUANG Q., HAUPTMANN A., BISK Y. & GAO J. (2022). KAT : A Knowledge Augmented Transformer for Vision-and-Language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 956–968, Seattle, United States : Association for Computational Linguistics.

HE K., ZHANG X., REN S. & SUN J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770–778.

HEO Y.-J., KIM E.-S., CHOI W. S. & ZHANG B.-T. (2022). Hypergraph Transformer : Weakly-supervised multi-hop reasoning for knowledge-based visual question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 373–390, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.29.

HU Z., ISCEN A., SUN C., WANG Z., CHANG K.-W., SUN Y., SCHMID C., ROSS D. A. & FATHI A. (2022). REVEAL : Retrieval-Augmented Visual-Language Pre-Training with Multi-Source Multimodal Knowledge Memory. arXiv :2212.05221 [cs], DOI : 10.48550/arXiv.2212.05221.

JOHNSON J., DOUZE M. & JÉGOU H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, **7**(3), 535–547. DOI : 10.1109/TBDATA.2019.2921572.

JOSHI M., CHOI E., WELD D. & ZETTLEMOYER L. (2017). TriviaQA : A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1601–1611, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1147.

KARPUKHIN V., OGUZ B., MIN S., LEWIS P., WU L., EDUNOV S., CHEN D. & YIH W.-T. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6769–6781, Online : Association for Computational Linguistics.

KHAN S., NASEER M., HAYAT M., ZAMIR S. W., KHAN F. S. & SHAH M. (2022). Transformers in vision : A survey. *ACM Comput. Surv.*, **54**(10s). DOI : 10.1145/3505244.

LERNER P., FERRET O. & GUINAUDEAU C. (2023). Multimodal inverse cloze task for knowledge-based visual question answering. In J. KAMPS, L. GOEURIOT, F. CRESTANI, M. MAISTRO, H. JOHO, B. DAVIS, C. GURRIN, U. KRUSCHWITZ & A. CAPUTO, Éds., *Advances in Information Retrieval*, p. 569–587, Cham : Springer Nature Switzerland. DOI : 10.1007/978-3-031-28244-7_36.

LERNER P., FERRET O., GUINAUDEAU C., LE BORGNE H., BESANÇON R., MORENO J. G. & LOVÓN MELGAREJO J. (2022). ViQuAE, a dataset for knowledge-based visual question answering about named entities. In *Proceedings of The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3477495.3531753.

LEWIS P., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., YIH W.-T., ROCKTÄSCHEL T., RIEDEL S. & KIELA D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems*, volume 33, p. 9459–9474 : Curran Associates, Inc.

LHOEST Q., VILLANOVA DEL MORAL A., JERNITE Y., THAKUR A., VON PLATEN P., PATIL S., CHAUMOND J., DRAME M., PLU J., TUNSTALL L., DAVISON J., ŠAŠKO M., CHHABLANI G., MALIK B., BRANDEIS S., LE SCAO T., SANH V., XU C., PATRY N., MCMILLAN-MAJOR A., SCHMID P., GUGGER S., DELANGUE C., MATUSSIÈRE T., DEBUT L., BEKMAN S., CISTAC

P., GOEHRINGER T., MUSTAR V., LAGUNAS F., RUSH A. & WOLF T. (2021). Datasets : A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 175–184, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics.

LIU Z., XIONG C., LV Y., LIU Z. & YU G. (2023). Universal vision-language dense retrieval : Learning a unified representation space for multi-modal retrieval. In *The Eleventh International Conference on Learning Representations*.

LOSHCHILOV I. & HUTTER F. (2019). Decoupled Weight Decay Regularization. *arXiv :1711.05101 [cs, math]*. arXiv : 1711.05101.

MA X., SUN K., PRADEEP R. & LIN J. (2021). A Replication Study of Dense Passage Retriever. *arXiv :2104.05740 [cs]*.

MARINO K., RASTEGARI M., FARHADI A. & MOTTAGHI R. (2019). OK-VQA : A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 3195–3204.

MOKADY R., HERTZ A. & BERMANO A. H. (2021). Clipcap : Clip prefix for image captioning. DOI : 10.48550/ARXIV.2111.09734.

PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J. & CHINTALA S. (2019). PyTorch : An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, **32**.

RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J. *et al.* (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, p. 8748–8763 : PMLR.

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.

RAMESH A., PAVLOV M., GOH G., GRAY S., VOSS C., RADFORD A., CHEN M. & SUTSKEVER I. (2021). Zero-shot text-to-image generation. In M. MEILA & T. ZHANG, Éds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 de *Proceedings of Machine Learning Research*, p. 8821–8831 : PMLR.

RASHTCHIAN C., YOUNG P., HODOSH M. & HOCKENMAIER J. (2010). Collecting image annotations using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, p. 139–147, USA : Association for Computational Linguistics.

SHAH S., MISHRA A., YADATI N. & TALUKDAR P. P. (2019). KVQA : Knowledge-Aware Visual Question Answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, p. 8876–8884.

SMUCKER M. D., ALLAN J. & CARTERETTE B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, p. 623–632, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/1321440.1321528.

SUN W., FAN Y., GUO J., ZHANG R. & CHENG X. (2022). Visual Named Entity Linking : A New Dataset and A Baseline. arXiv :2211.04872 [cs].

WANG J., GONG T., ZENG Z., SUN C. & YAN Y. (2022). C3CMR : Cross-Modality Cross-Instance Contrastive Learning for Cross-Media Retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, p. 4300–4308, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3503161.3548263.

WANG Z., NG P., MA X., NALLAPATI R. & XIANG B. (2019). Multi-passage BERT : A Globally Normalized BERT Model for Open-domain 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. 5878–5882, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1599.

WESTON J., CHOPRA S. & BORDES A. (2014). Memory networks. DOI : 10.48550/ARXIV.1410.3916.

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., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). HuggingFace's Transformers : State-of-the-art Natural Language Processing. *arXiv :1910.03771 [cs]*.

WOLFE R. & CALISKAN A. (2022). Contrastive Visual Semantic Pretraining Magnifies the Semantics of Natural Language Representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 3050–3061.

ZAMANI H., DIAZ F., DEHGHANI M., METZLER D. & BENDERSKY M. (2022). Retrieval-Enhanced Machine Learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, p. 2875–2886, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3477495.3531722.

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

Adaptation de domaine pour la recherche dense par annotation automatique

Minghan Li[1] Eric Gaussier[1]
(1) Univ. Grenoble Alpes, CNRS, LIG, Bâtiment IMAG - 700 avenue Centrale, 38000 Grenoble, France
minghan.li@univ-grenoble-alpes.fr, eric.gaussier@imag.fr

RÉSUMÉ

Bien que la recherche d'information neuronale ait connu des améliorations, les modèles de recherche dense ont une capacité de généralisation à de nouveaux domaines limitée, contrairement aux modèles basés sur l'interaction. Les approches d'apprentissage adversarial et de génération de requêtes n'ont pas résolu ce problème. Cet article propose une approche d'auto-supervision utilisant des étiquettes de pseudo-pertinence automatiquement générées pour le domaine cible. Le modèle T53B est utilisé pour réordonner une liste de documents fournie par BM25 afin d'obtenir une annotation des exemples positifs. L'extraction des exemples négatifs est effectuée en explorant différentes stratégies. Les expériences montrent que cette approche aide le modèle dense sur le domaine cible et améliore l'approche de génération de requêtes GPL.

ABSTRACT

Domain adaptation with pseudo-relevance labeling for dense retrieval.

Although neural information retrieval has witnessed great improvements, recent works showed that the generalization ability of dense retrieval models on target domains with different distributions is limited, which contrasts with the results obtained with interaction-based models. To address this issue, researchers have resorted to adversarial learning and query generation approaches ; both approaches nevertheless resulted in limited improvements. In this paper, we propose to use a self-supervision approach in which pseudo-relevance labels are automatically generated on the target domain. To do so, we use the interaction-based model T53B to re-rank the BM25 list on target domain for pseudo positive labeling. Since negative mining is vital, we carefully design it with investigating different negative mining strategies. Our experiments reveal that the proposed pseudo-relevance labeling approach helps the dense retrieval model on target domain and improves the state-of-the-art query generation approach GPL when they are fine-tuned on the generated data.

MOTS-CLÉS : Adaptation de domaine, Apprentissage auto-supervisé, Recherche d'information neuronale.

KEYWORDS: Dense Retrieval, Domain Adaptation, Self-Supervised Learning, Neural IR.

1 Introduction

La recherche d'information (RI) joue un rôle crucial dans notre vie quotidienne en raison de l'explosion des données. Les approches traditionnelles de RI telles que BM25 (40) calculent une similarité entre une requête et un document uniquement sur la base des termes communs aux deux. En tant que telles, elles ne peuvent pas gérer la correspondance sémantique entre différentes formes de surface.

La recherche d'informations neuronale, avec l'avènement des réseaux de neurones profonds, a considérablement amélioré les systèmes de RI grâce à des modèles capables de capturer la sémantique de chaque terme et de les comparer même si leur forme de surface diffère. Un modèle populaire à la fois en traitement du langage naturel (NLP) et en RI est BERT (4), qui est basé sur des transformateurs (47) et est pré-entraîné sur de grandes collections ; BERT peut être utilisé sur une variété de tâches secondaires par adaptation (en anglais *fine-tuning*).

Les modèles de RI neuronale peuvent être classés en deux catégories (11) : les approches basées sur l'interaction et les approches basées sur la représentation (également appelées recherche dense). Les modèles basés sur l'interaction ont montré des performances moyennes supérieures à celles des modèles de recherche dense ; les modèles de recherche dense sont toutefois plus rapides que les modèles basés sur l'interaction, car les représentations des documents peuvent être générées et stockées à l'avance, et sont préférables si l'on a besoin de déployer un modèle à grande échelle. Cela étant dit, des études récentes comme BEIR (44) ont montré que les modèles de recherche dense entraînés sur un domaine source généralisent moins bien que les modèles traditionnels tels que BM25 et les modèles basés sur l'interaction sur des ensembles de données hors distribution (OOD pour *Out Of Distribution*). Bien que l'entraînement sur des ensembles de données cibles avec des étiquettes de référence soit un processus standard, l'annotation requise peut être à la fois longue et coûteuse, de sorte que cette approche peut ne pas être applicable à de nombreuses utilisations réelles. Il est donc important de traiter le problème dans des scénarios OOD pour la recherche dense.

L'un des objectifs de l'adaptation de domaine (53; 50) est de permettre à un modèle entraîné sur un domaine appelé le domaine source de bien fonctionner sur un autre domaine appelé le domaine cible sans utiliser d'étiquettes humaines sur ce dernier. Récemment, différentes techniques d'adaptation de domaine pour la recherche dense ont été proposées. La généralisation de domaine basée sur la génération de données est l'une des approches (50) qui a été suivie dans (26) grâce à un modèle appelé QGen qui génère des requêtes pour le domaine cible en utilisant un générateur de requêtes entraîné sur le domaine source. Dans la même lignée, GPL (52) utilise des exemples négatifs difficiles et la distillation de connaissances et obtient des résultats de pointe sur un certain nombre d'ensembles de données BEIR. Cependant, les requêtes créées sont synthétiques et peuvent ne pas ressembler à de véritables requêtes cibles. Une autre approche populaire et largement utilisée est basée sur l'apprentissage adversarial (50). Très récemment, Xin et al. (54) ont proposé un modèle appelé MoDIR qui entraîne de manière adverse un encodeur de recherche dense pour apprendre des représentations invariantes aux domaines. Cependant, un tel objectif d'apprentissage peut produire un espace de plongement (en anglais *embedding*) de faible qualité et entraîner des performances instables (52; 14).

Dans cet article, nous proposons une approche dénommée DoDress (pour **Do**main generalization for **D**ense **re**trieval through **s**elf-**s**upervision) qui cherche d'abord à construire des annotations de pseudo-pertinence [1] sur le domaine cible en utilisant des modèles basés sur l'interaction uniquement entraînés sur le domaine source tels que T53B (34). La raison d'utiliser des modèles basés sur l'interaction dans ce contexte réside dans le fait que ces modèles ont montré un comportement relativement bon sur les ensembles de données OOD (44). Notez que le modèle T53B lourd n'est utilisé que pour produire des étiquettes de pseudo-relation avant l'entraînement du modèle de recherche dense afin que l'approche globale reste efficace pendant la recherche en ligne. Cette méthode élimine le besoin d'annotations humaines et permet au modèle d'utiliser de véritables requêtes et documents du domaine cible.

Une des difficultés dans l'annotation automartique est d'obtenir des exemples négatifs de qualité.

1. Nous utilisons ce terme pour rendre compte du fait que certaines de ces annotations sont erronées.

Nous étudions pour cela dans cet article différentes stratégies : l'échantillonnage négatif aléatoire global, les exemples négatifs "durs" de BM25 et les exemples négatifs "durs" de SimANS (16).

Notre contribution réside dans l'étude et la combinaison de différentes approches existantes pour l'annotation automatique dans un domaine cible qui conduit *n fine* à une méthode qui améliore l'état de l'art pour l'adaptation de domaine de modèles denses.

2 Travaux connexes

Wang *et al.* (50) présentent un article de synthèse sur la généralisation de domaine pour les domaines non vus. La généralisation ou l'adaptation de domaine peut être catégorisée en trois groupes : manipulation de données, apprentissage de représentation et stratégie d'apprentissage. Il existe deux types de techniques dans le premier groupe : l'augmentation de données (35; 45; 43; 48) qui est couramment utilisée dans les données d'images (par exemple, en modifiant la localisation, le texte des objets et en ajoutant du bruit aléatoire) et la génération de données (38; 36; 57) qui utilise certains modèles pour générer de nouvelles données pour entraîner un modèle. Le groupe d'apprentissage de représentation comprend l'apprentissage de représentation invariante de domaine (par exemple, l'apprentissage adversarial de domaine) (1; 7; 31) et les méthodes de désentrelacement de fonctionnalités (21; 32; 24). Le troisième groupe comporte plusieurs catégories, comme l'apprentissage en ensemble (28; 6), l'apprentissage méta (20; 5) et les approches basées sur l'apprentissage auto-supervisé (par exemple, la résolution de puzzles de type jigsaw) (3; 13).

Des stratégies similaires, telles que la généralisation de domaine ou l'apprentissage par transfert, sont avancées par les chercheurs pour la recherche d'informations. Une stratégie similaire à celle adoptée dans cette étude est décrite dans (30), qui effectue une évaluation systématique de la capacité de transfert des modèles de classement neuronaux basés sur BERT. Les auteurs utilisent également BM25 pour générer des étiquettes de pseudo-pertinence. Ils ne se concentrent cependant pas sur les modèles de recherche denses qui sont connus pour nécessiter des méthodes d'entraînement complexes et une grande quantité de données dans une situation distincte (8). De plus, l'utilisation uniquement de BM25 pour obtenir des étiquettes de pseudo-pertinence pourrait être une solution faible. Pour les modèles basés sur l'interaction (19) ou l'apprentissage d'embeddings de phrases (10; 51), certaines publications suggèrent des techniques auto-supervisées. Ces méthodes sont fréquemment utilisées pour la pré-formation, mais elles ne se concentrent pas explicitement sur la généralisation de domaine (52). Ma *et al.* (26) propose QGen, une approche de génération d'apprentissage zéro pour la première étape de la recherche dense de passages qui utilise la génération de questions synthétiques, permettant la construction de paires de pertinence question-passage arbitrairement grandes mais bruyantes qui sont spécifiques au domaine, dans le but de surmonter le défi que les modèles de recherche neuronaux ont besoin d'un grand ensemble d'entraînement supervisé pour surpasser les approches conventionnelles basées sur les termes. Les documents utilisés pour générer les requêtes sont considérés comme des passages positifs et les autres instances dans le lot sont considérées comme négatives. En parallèle, Liang *et al.* (23) examinent l'architecture de recherche de passages denses à deux tours. Étant donné que les données étiquetées peuvent être difficiles à obtenir et que les modèles de recherche neuronaux ont besoin d'une grande quantité de données pour être entraînés, ils suggèrent également d'utiliser des requêtes synthétiques produites par un grand modèle de séquence à séquence (seq2seq) pour l'adaptation de domaine non supervisée. Ces deux articles montrent l'efficacité de l'approche de génération de requêtes, qui est également utilisée dans le modèle GPL (52). GPL

s'appuie sur un encodeur-décodeur T5 pré-entraîné (37) pour générer des requêtes à partir de passages d'entrée. Les passages d'entrée sont considérés comme des passages positifs tandis que les passages similaires récupérés à l'aide d'un modèle de recherche dense existant sont constitués de passages négatifs (difficiles). La perte Margin-MSE (12) est utilisée comme distillation de connaissances pour enseigner au modèle de recherche dense à apprendre à partir d'un modèle basé sur l'interaction. Les résultats expérimentaux montrent une efficacité de pointe sur plusieurs collections BEIR (44).

Les chercheurs ont également exploré des stratégies alternatives pour l'adaptation de domaine des modèles de recherche dense. Xin *et al.* (54) a proposé une approche d'apprentissage de représentation invariante de domaine adversaire avec une méthode d'impulsion pour l'apprentissage de classifier de domaine qui distingue les domaines source et cible. L'encodeur de recherche dense est ensuite formé de manière adversaire pour apprendre des représentations invariantes de domaine. Une file d'attente à impulsion qui enregistre des embeddings de plusieurs lots précédents est utilisée afin de trouver un équilibre entre précision et efficacité (54). Cette approche est utilisée sur un modèle ANCE entraîné (55). Les résultats varient d'un ensemble de données à l'autre, avec parfois des améliorations importantes et parfois des gains ou pertes marginales. Karouzos *et al.* (14) a proposé UDALM pour l'adaptation de domaine pour la classification de sentiment à travers l'apprentissage multi-tâche. Il apprend simultanément l'objectif de la tâche de modélisation de langage masquée (MLM) sur le domaine cible et la tâche à partir des données étiquetées source. Cependant, cette stratégie n'a pas été conçue pour la recherche dense et, comme mentionné dans (52), elle ne fonctionne pas bien pour la recherche dense.

Dans cet article, nous proposons de faire l'adaptation de domaine pour la recherche dense par auto-supervision par étiquetage de pseudo-relevance. Nous appliquons le modèle d'interaction T53B de pointe et généralisable au domaine (34) pour l'étiquetage de pseudo-positif. Ce modèle peut produire des étiquettes de pseudo-relevance plus précises, où les documents classés en haut sont considérés comme pertinents pour une requête donnée. De plus, différentes stratégies d'échantillonnage négatives sont étudiées, en particulier avec les négatifs durs de SimANS (16) échantillonnés à partir de la liste de recherche des modèles DR actuels, pour améliorer l'efficacité du modèle, après la formation avec les données pseudo-étiquetées générées.

3 Contexte

Recherche dense La recherche dense (15; 54) vise à encoder à la fois les requêtes et les documents dans un espace de faible dimension à l'aide d'un encodeur g, généralement un modèle de type BERT. Le score de pertinence (RSV) d'une requête et d'un document est ensuite calculée à l'aide d'une fonction de similarité simple dans l'espace de faible dimension :

$$RSV(q,d)_{DR} = g(q) \cdot g(d) \quad (or\ RSV(q,d)_{DR} = cos(g(q), g(d)))\,, \tag{1}$$

où $g(q)$ (resp. $g(d)$) représente l'encodage de la requête (resp. du document). Cela permet une recherche rapide en utilisant par exemple la méthode proposée par Xiong *et al.* (55).

BM25 BM25 est un algorithme standard en RI basé sur la correspondance de termes. Le RSV d'un document par rapport à une requête est donné par :

$$RSV(q,d)_{BM25} = \sum_{w \in q \cap d} IDF(w) \cdot \frac{tf_w}{k_1 \cdot (1 - b + b \cdot \frac{l_d}{l_{avg}}) + tf_w}, \tag{2}$$

où $IDF(w)$ est la fréquence inverse des documents, l_d est la longueur du document d, l_{avg} est la longueur moyenne des documents dans l'ensemble de données, et k_1 et b sont deux hyperparamètres.

T53B T5 (37) est un modèle qui a montré son efficacité dans diverses tâches du traitement automatique des langues. Nogueira *et al.* (34) ont proposé d'utiliser T5 en tant que modèle basé sur l'interaction pour la recherche d'informations en se basant sur la représentation d'entrée suivante :

Query : [q] Document : [d] Relevant : true or false

où $[q]$ et $[d]$ sont remplacés par les textes de la requête et du document. Pendant l'entraînement, le modèle T5 apprend à générer le mot "true" lorsque le document est pertinent pour la requête, et le mot "false" lorsqu'il ne l'est pas. Le score de pertinence pour l'inférence est ensuite déterminé par la probabilité de produire "true" (34) :

$$RSV(q,d)_{T5} = softmax(Z_{true}) = \frac{e^{Z_{true}}}{e^{Z_{true}} + e^{Z_{false}}}, \tag{3}$$

où Z_{true} et Z_{false} sont les logits des tokens de sortie.

4 DoDress : annotation automatique de pertinence

Nous proposons simplement ici de considérer les k meilleurs documents, obtenus avec la combinaison BM25&T53B dans laquelle T53B ré-ordonne les documents fournis par BM25, comme pertinents. k est un hyperparamètre qui peut être ajusté en fonction de différentes informations, telles que le nombre de requêtes et de documents disponibles. Pour chaque paire (requête, document pertinent) obtenue, nous cherchons à extraire de la collection m documents non pertinents pour la requête. Ainsi, pour chaque requête, $k \times m$ triplets (requête, document pertinent, document non pertinent) sont consituées. Les blocs verts dans les Figures 1 et 2 représentent ces triplets qui constituent les données d'entraînement sur le domaine cible.

Une stratégie simple d'extraction de documents non pertinents consiste à un échantillonnage aléatoire global de la collection excluant les documents jugés pertinents. Toutefois, les modèles de recherche denses nécessitent des stratégies d'apprentissage complexes pour être performants (9) et l'approche précédente ne garantit pas que les documents non pertinents obtnues soient suffisamment informatifs. Un des défis clé pour la recherche dense est en effet de construire des instances négatives appropriées pour l'apprentissage (15). Une solution possible ici est d'utiliser des documents non pertinents proches des documents pertinents en termes de recherche obtenus par BM25. Nous échantillonons $k \times m$ documents parmi les documents de haut rang de BM25, en excluant bien sûr les k documents jugés pertinents après ré-ordonnancement par T53B, et les considérons comme non pertinents. Cette approche est illustrée dans la Figure 1 : les documents non pertinents sont échantillonnés au hasard

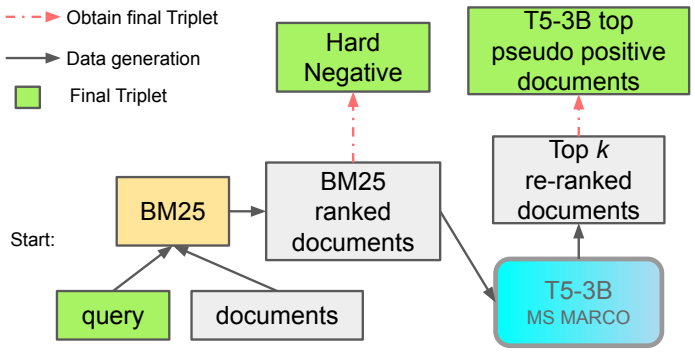


FIGURE 1 – Processus global d'annotation automatique avec échantillonnage négatif sur les résultats de BM25.

dans la liste de haut rang de BM25 alors que les documents pertinents correspondent aux k prem :iers documents obtenus après ré-ordonnancement par T53B.

L'approche précédente d'annotation est basée sur les instances les mieux classées et sur des exemples négatifs aléatoires globaux ou des exemples négatifs difficiles de BM25. Bien que les résultats soient globalement bons, des chercheurs ont récemment montré (16) que les stratégies d'échantillonnage négatif existantes souffrent du problème de faux négatifs ou d'informations non pertinentes, et ils montrent que les exemples négatifs classés autour des exemples positifs (par exemple, les scores BM25 ou les scores de recherche dense) sont généralement plus informatifs et moins susceptibles d'être des faux négatifs. Dans cet article, nous utilisons SimANS (16) comme illustré dans la figure 2. SimANS permet de sélectionner les documents non pertinents dans les classements des modèles denses D-BERT et GPL (voir Section **??**). À noter que les documents non pertinents classés autour de documents jugés pertinents ne sont pas forcément en tête de liste car les classements de D-BERT et GPL diffèrent de ceux de T53B. Toutefois, afin de ne pas sélectionner des documents trop mal classés, nous nous concentrons sur les Top500 documents fournis par D-BERT et GPL respectivement. Cela signifie que certains documents jugés pertinents sont susceptibles de ne plus être considérés s'ils n'appartiennet pas à ce Top500.

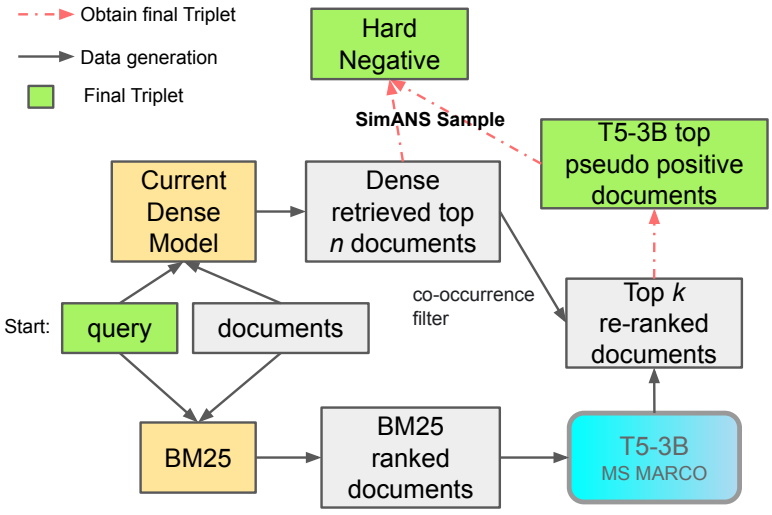


FIGURE 2 – Processus global d'annotation avec SimANS.

4.1 Combinaison avec GPL

Comme mentionné précédemment, les approches QGen et GPL s'appuient toutes deux sur un générateur de requêtes pour générer des pseudo-requêtes afin de construire un modèle de recherche dense. Nous proposons ici de construire un tel modèle sur les triplets de pseudo-pertinence décrits précédememnt et obtenus à partir des requêtes fournies par QGen ou GPL. Nous pensons qu'il est possible de tirer profit de cette formation supplémentaire sur la collection cible, car les pseudo-requêtes et les étiquettes de pseudo-pertinence reposent sur des sources d'informations différentes et sont complémentaires l'une de l'autre. Comme nous le verrons dans la section expérimentale, cette combinaison améliore effectivement l'approche de génération de pseudo-requêtes.

4.2 Fonction de coût

Dans cet article, nous nous appuyons sur la perte paire RankNet (2; 22) pour entraîner un modèle de recherche dense en utilisant les triplets générés ci-dessus, définis par :

$$\mathcal{L}(q, d^+, d^-; \Theta) = -\log(\sigma(S_{q,d^+} - S_{q,d^-})), \tag{4}$$

où q est une requête, (d^+, d^-) est une paire de documents d'entraînement (positif, négatif) pour q, σ est la fonction sigmoïde, Θ représente les paramètres du modèle de recherche dense, et $S_{q,d}$ est le score fourni par le modèle pour le document d par rapport à la requête q.

5 Expérimentation

5.1 Ensembles de données

Le jeu de données MS MARCO (33) est utilisé comme données de domaine source. Nous voulons expérimenter dans un scénario extrême où aucune requête de test ne peut être vue, même sans étiquettes humaines. Cela signifie que nous devons générer les données d'entraînement avec les requêtes d'entraînement qui ne sont pas dans l'ensemble de test. Pour ce faire, nous expérimentons sur 3 ensembles de données de domaine cible du benchmark BEIR (44) : FiQA, ensemble de questions-réponses sur la finance (27) qui contient 6000 requêtes d'entraînement, BioASQ, ensemble de questions-réponses dans le domaine biomédical (46) (suivant (52), les documents non pertinents sont éliminés au hasard pour ne conserver qu'un million de documents) qui contient 3243 requêtes d'entraînement provenant de la collection originale [2], et Robust04, ensemble documents d'actualités (49) qui contient 250 requêtes. Différents sujets et tâches sont couverts par ces ensembles. Pour Robust04, nous sélectionnons les 100 premières requêtes comme ensemble d'entraînement et de développement ; les 150 dernières requêtes sont utilisées comme ensemble de test.

2. http://participants-area.bioasq.org/Tasks/8b/trainingDataset/

5.2 Protocole expérimental

Notre implémentation est basée sur le cadre open-source Matchmaker [3] avec pooling moyen et évaluation dense [4] en utilisant la précision mixte automatique (29). Sur la collection cible, les triplets d'entraînement sont générés selon les approches décrites ci-dessus. Pour l'adaptation de domaine, nous utilisons deux modèles denses, D-BERT et GPL. D-BERT correspond au modèle DistilBERT (41) avec 6 couches. GPL est d'abord entraîné sur les pseudo-requêtes cibles qu'il génère et les documents associés avant d'être entraîné sur les triplets cibles. Le modèle T5 utilisé est la version 3B qui est entraînée sur l'ensemble de données de classement de passage MS MARCO [5]. Le *cross-encoder* MiniLM utilisé est la version *ms-marco-MiniLM-L-6-v2* [6] de *sentence transformers* (39).

Pour construire l'ensemble d'entraînement, nous sélectionnons le nombre k de documents principaux à considérer comme pertinents en fonction du nombre de requêtes (pour générer suffisamment de paires) et de documents (un grand nombre de documents permettant d'échantillonner plus de documents négatifs). Pour chaque document pertinent, nous sélectionnons m documents de la collection cible qui ne figurent pas dans la liste des k premiers documents de la requête selon les différentes stratégies d'échantillonnage négatif présentées précédemment. Ces documents sont considérés comme non pertinents. Le Tableau 1 affiche le nombre de requêtes, la valeur sélectionnée pour k (entre parenthèses) et le nombre m de documents non pertinents par document pertinent. Par exemple, pour BioASQ, le nombre de triplets dans l'ensemble d'entraînement est de $3193 \times 2 \times 15 = 95790$. À la fin, chaque ensemble de données a un nombre de triplets dans l'ensemble d'entraînement compris entre 50000 et 100000. Nous construisons également un ensemble de développement pour sélectionner les hyperparamètres des modèles sur chaque collection. Pour chaque requête de l'ensemble de développement, les 10 premiers documents sont considérés comme pertinents et 90 documents sélectionnés au hasard comme non pertinents. Ce choix est dicté par le fait que nous avons besoin d'un nombre suffisant de documents pertinents à des fins d'évaluation et que nous avons un nombre limité de requêtes pour l'ensemble de développement. Cependant, pour contrebalancer le risque de considérer comme pertinents des documents qui ne le sont pas en réalité, les deux premiers documents sont étiquetés "2" et les huit suivants comme "1". Les documents non pertinents sont étiquetés "0", ce qui conduit à des jugements de pertinence à 3 niveaux pour chaque ensemble de données. Le meilleur modèle est sauvegardé en fonction du score NDCG@10 sur l'ensemble de développement évalué toutes les 1 000 étapes.

Suivant (52), une longueur de séquence maximale de 350 et une similarité par produit scalaire sont utilisées. Pour tous les ensembles de données, nous utilisons une taille de lot de 8, ce qui signifie 8 paires positives-négatives, et un taux d'apprentissage de 2e-6 avec un optimiseur Adam pour 10 000 étapes d'entraînement. Un schéma LR cosinus (25) est également utilisé pour la décroissance du taux d'apprentissage.

Pour SimANS, les hyperparamètres a et b sont fixés à 0,5 et 0 respectivement pour toutes les expériences.

3. https://github.com/sebastian-hofstaetter/matchmaker
4. https://github.com/UKPLab/gpl/blob/main/gpl/toolkit/evaluation.py
5. https://huggingface.co/castorini/monot5-3b-msmarco
6. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

TABLE 1 – Nombre de requêtes, de documents et de documents pertinents et non pertinents par requête pour chaque collection. Les nombres entre parenthèses correspondent aux requêtes utilisées pour l'entraînement.

dataset	#requêtes	#docs	k	m
FiQA	6000 (5960)	57K	1	10
BioASQ	3243 (3193)	1M	2	15
Robust04	100 (90)	528K	15	67

5.3 Modèles *baseline*

Conformément à (52), nous comparons les approches proposées avec des modèles sans apprentissage, avec des approches de pré-entraînement et avec les approches récentes les plus performantes pour l'adaptation de domaine. Les résultats des modèles basés sur l'interaction sont également présentés.

5.3.1 Modèles sans apprentissage

Les modèles de référence sans apprentissage comprennent BM25 basé sur Anserini (56) avec des paramètres par défaut qui obtient les 100 meilleurs documents pour chaque requête et ne nécessite pas d'être entraîné (ces listes de classement BM25 sont ensuite utilisées pour générer des données d'apprentissage de pseudo-pertinence dans cet article) et le modèle de recherche dense D-BERT entraîné uniquement sur la collection source, la fonction de coût MarginMSE et le modèle *ms-marco-MiniLM-L-6-v2* pris comme modèle enseignant (il est ensuite utilisé comme point de départ pour l'adaptation de domaine).

5.3.2 Modèles basés sur le pré-entraînement

Nous comparons avec SimCSE (10), ICT (19) et TSDAE (51). Ces modèles sont tous d'abord pré-entraînés de manière auto-supervisée sur l'ensemble de données cible, puis affinés sur MS MARCO.

5.3.3 Approches d'adaptation de domaine

Nous comparons ici quatre approches SOTA récentes : MoDIR (54), qui repose sur ANCE (55) et utilise un entraînement adversarial, UDALM (14), qui repose sur l'apprentissage multi-tâches, et QGen (26) et GPL (52), qui sont des approches basées sur la génération de requêtes.

De plus, nous utilisons les modèles basés sur l'interaction BM25+CE et BM25+T53B, qui peuvent être considérés comme des baselines solides en raison du bon comportement des modèles basés sur l'interaction dans les contextes OOD (44), mais qui restent néanmoins inefficaces lors de l'inférence. Ces modèles réordonnent la liste des 100 premiers résultats renvoyés par BM25, en utilisant respectivement les *corss-encoders ms-marco-MiniLM-L-6-v2* et T53B.

5.4 Résultats et analyse

Les tableaux 3, 4 et 5 affichent les résultats obtenus avec les différents modèles et approches. Les résultats rapportés pour BM25+CE, UDALM, MoDIR, SimCSE, ICT, TDSAE, QGen et TSDAE+GPL proviennent de (52). Comme nous testons le dataset Robust04 sur les 150 dernières requêtes, pour BM25+CE, GPL et TSDAE+GPL, nous chargeons les points de contrôle entraînés de D-BERT et de Wang *et al.* (52) [7], et les évaluons sur les 150 dernières requêtes. La notation "DoDress-BM25 (D-BERT)" (respectivement "DoDress-T53B (D-BERT)") correspond au modèle de recherche dense D-BERT pré-entraîné sur MS MARCO et affiné sur les données cibles en utilisant les documents pertinents obtenus par BM25 (respectivement en utilisant BM25+T5). La notation (GPL) signifie la même chose pour GPL, qui est d'abord entraîné sur les pseudo-requêtes cibles qu'il génère et les documents associés avant d'être entraîné sur les triplets cibles.

Nous analysons les résultats en répondant à trois questions de recherche.

RQ1 Les top positifs de BM25+T53B aident-ils à la généralisation de domaine pour les modèles de recherche dense ?

À partir du Tableau 3, on constate que DoDress-T53B (D-BERT) améliore D-BERT sur le jeu de données FiQA avec les trois stratégies d'échantillonnage négatives, et que DoDress-T53B (GPL) montre une tendance similaire par rapport à GPL. Sur le jeu de données Robust04, à partir du Tableau 4, nous observons des tendances similaires pour DoDress-T53B (D-BERT) et DoDress-T53B (GPL). Plus précisément, DoDress-T53B (GPL) avec les trois stratégies d'échantillonnage permet d'améliorer GPL et TSDAE + GPL. Avec la stratégie de recherche de négatifs SimANS, DoDress-T53B (D-BERT) montre une amélioration de 11,5% $((43,6 - 39,1) \div 39,1)$ par rapport à D-BERT, et DoDress-T53B (GPL) montre une amélioration de 8,6% par rapport à GPL. Toutefois, à partir du Tableau 5, l'approche avec la stratégie globale d'échantillonnage négative aléatoire échoue, tandis que l'approche proposée avec les deux autres stratégies d'échantillonnage négatives améliore respectivement le modèle de recherche dense D-BERT et GPL.

Ces approches montrent que l'approche d'annotation automatique proposée peut aider les modèles de recherche dense à généraliser vers de nouveaux domaines, et que le choix de la stratégie d'échantillonnage négative est important pour cela.

RQ2 Quel est l'impact des différentes stratégies d'échantillonnage négatives et laquelle est la meilleure ?

Dans les trois tableaux, nous observons une tendance globalement ascendante des trois différentes stratégies d'échantillonnage négatives. Bien que la méthode négative globale aléatoire améliore les modèles de recherche dense sur FiQA et Robust04, elle échoue sur le jeu de données BioASQ, pour lequel la distinction entre documents pertinents et non pertinents semble plus difficile.

En ce qui concerne les stratégies d'échantillonnage de négatifs BM25 et SimANS, elles montrent de meilleures performances que la stratégie de négatifs aléatoires globaux, et améliorent D-BERT et GPL sur les trois ensembles de données. Ces résultats montrent que l'échantillonnage de négatifs difficiles est important pour l'annotation automatique en RI.

L'approche proposée avec l'échantillonnage de négatifs difficiles SimANS donne systématiquement les meilleurs résultats sur tous les ensembles de données, meilleure que les stratégies de négatifs aléatoires globaux et de négatifs BM25, montrant qu'une meilleure stratégie d'échantillonnage de

7. https://huggingface.co/GPL

négatifs peut également améliorer davantage l'approche proposée.

RQ3 Quel est l'effet de l'approche d'annotation proposée avec l'échantillonnage de négatifs SimANS par rapport aux modèles de référence ?

Nous analysons maintenant l'approche proposée avec la stratégie d'échantillonnage de négatifs SimANS en la comparant aux modèles de référence.

BM25 est un algorithme de recherche standard considéré comme un modèle sans apprentissage. Bien qu'il soit extrêmement simple par rapport aux approches neurales de RI récentes, c'est une baseline solide et performante sur de nouveaux domaines. Surtout sur BioASQ, l'approche BM25 surpasse même l'approche de réordonnancement BM25 + CE sur un nouveau domaine. Notre approche proposée le surpasse sur FiQA et Robust04. En particulier, DoDress-T53B (GPL) avec SimANS est la seule approche qui surpasse BM25 sur Robust04, où GPL et TSDAE + GPL échouent. Sur BioASQ, tous les modèles denses sont inférieurs à BM25, tandis que DoDress-T53B (GPL) est le meilleur parmi eux.

Pour UDALM, MoDIR (ANCE) et les trois approches basées sur le pré-entraînement SimCSE, ICT et TSDAE, sur Robust04 nous n'avons pas les points de contrôle entraînés pour évaluer les 150 dernières requêtes de test, et nous montrons les résultats sur FiQA et BioASQ. DoDress-T53B (D-BERT) et DoDress-T53B (GPL) avec des négatifs obtenus par SimANS surclassent systématiquement tous les autres modèles. La meilleure baseline sur FiQA est MoDIR (ANCE) avec 29,6, tandis que nos DoDress-T53B (D-BERT) et DoDress-T53B (GPL) sont respectivement à 31,0 et 34,9. Sur BioASQ, la meilleure baseline parmi ces modèles est TSDAE avec 55,5, tandis que nos DoDress-T53B (D-BERT) et DoDress-T53B (GPL) sont à 60,6 et 65,3 respectivement, le dernier ayant une amélioration de 17,7%.

Les modèles de génération sont des modèles d'état de l'art précédents, principalement basés sur les pseudo-requêtes générées à partir de documents considérés comme pertinents. Dans notre approche, nous prenons également en compte les vraies requêtes. Dans le Tableau 3, nous voyons que TSDAE + GPL est le meilleur des modèles de base, avec un score de 34,4, mieux que les 32,8 de GPL. L'approche proposée, DoDress-T53B (GPL), obtient 34,9, ce qui est plus élevé que ces scores. Cependant, sur Robust04, dans le Tableau 4, TSDAE + GPL est moins bon que GPL : 40,7 et 41,9 respectivement. Notre approche proposée avec des négatifs difficiles échantillonnés avec SimANS, DoDress-T53B (D-BERT) et DoDress-T53B (GPL), les surpasse tous les deux, montrant ainsi l'efficacité de l'approche. Sur le jeu de données BioASQ, DoDress-T53B (GPL) obtient de meilleurs résultats que le meilleur modèle GPL dans les modèles de base.

RQ4 L'échantillonnage de documents non pertinents dans la liste fournie par un modèle dense fonctionne-t-il mieux et SimANS peut-il encore l'améliorer ?

Nous voulons voir si échantillonner les négatifs dans la liste du modèle de recherche dense conduit à de meilleurs résultats que l'échantillonnage dans la liste BM25. Nous menons donc une expérience supplémentaire en utilisant un échantillonnage négatif aléatoire à partir de la liste de recherche du modèle dense que nous cherchons à construire. Les résultats sont présentés dans le Tableau 2. Nous pouvons voir que l'échantillonnage des négatifs à partir de la liste fournie par le modèle D-BERT en cours de création est meilleur que celui à partir de la liste BM25. Cela peut s'expliquer par le fait que les négatifs provenant de la liste D-BERT sont plus ambigus pour ce modèle et donc plus informatifs pour son entraînement et l'adaptation à un nouveau domaine.

En outre, nous voulons voir si l'échantillonnage de SimANS peut encore améliorer les résultats de

TABLE 2 – Résultats de DoDress-BM25 (D-BERT) sur Robust04 avec différentes sources d'échantillonnage aléatoire de négatifs.

Échantillonnage aléatoire à partir de	nDCG@10 (%)
Liste supérieure BM25	41.6
Liste supérieure GPL	42.6

recherche. Dans le tableau 4, nous constatons que DoDress-BM25 (D-BERT) utilisant l'approche d'échantillonnage SimANS obtient 43,6, tandis que dans le tableau 2, il est de 42,6, ce qui montre que SimANS peut encore améliorer le résultat en échantillonnant des négatifs plus ambigus que l'échantillonnage aléatoire à partir de la liste de classement supérieure du modèle dense actuel.

En conclusion, les résultats ci-dessus démontrent l'efficacité de l'approche proposée combinant différents modèles. Ils confirment également l'importance du choix des documents non pertinents et le bon comportement de l'approche SimANS dans ce cadre.

TABLE 3 – Résultat d'adaptation de domaine de FiQA (en utilisant uniquement les requêtes d'entraînement).

modèle	nDCG@10 (%)
Modèles sans adaptation	
D-BERT	26.7
BM25 (Anserini)	23.6
Re-Ranking avec des Cross-Encoders (limite supérieure)	
BM25 + CE	33.1
BM25 + T53B	39.2
Méthodes précédentes d'adaptation de domaine	
UDALM	23.3
MoDIR (ANCE)	29.6
Pré-entraînement basé : Cible → D-BERT	
SimCSE	26.7
ICT	27.0
TSDAE	29.3
Basé sur la génération (SOTA précédent)	
QGen	28.7
GPL	32.8
TSDAE + GPL	34.4
Proposée : T53B, Négatifs Aléatoires Globaux	
DoDress-T53B (D-BERT)	27.3
DoDress-T53B (GPL)	33.0
Proposé : T53B, Négatifs durs BM25	
DoDress-BM25 (D-BERT)	30.4
DoDress-BM25 (GPL)	34.2
Proposé : T53B, Négatifs durs SimANS	
DoDress-T53B (D-BERT)	**31.0**
DoDress-T53B (GPL)	**34.9**

6 Conclusion

Nous avons étudié dans cet article s'il est possible d'annoter automatiquement des documents dans un domaine cible de façon à y déployer un modèle de RI dense. Notre étude révèle que cette approche fonctionne bien lorsque les annotations sont générées à l'aide d'un modèle T53B ré-ordonnant les documents obtenus par BM25, et qu'elle aide à améliorer les résultats de généralisation du modèle GPL qui utilise également des requêtes générées et des documents pertinents associés sur la collection

TABLE 4 – Résultats d'adaptation de domaine de Robust04 (l'ensemble d'entraînement et de développement utilise les 100 premières requêtes, l'ensemble de test est constitué des 150 dernières requêtes).

modèle	nDCG@10 (%)
Modèles sans adaptation	
D-BERT	39.1
BM25 (Anserini)	44.4
Re-Ranking avec des Cross-Encoders (limite supérieure)	
BM25 + CE	45.8
BM25 + T53B	51.8
Basé sur la génération (SOTA précédent)	
GPL	41.9
TSDAE + GPL	40.7
Proposée : T53B, Négatifs Aléatoires Globaux	
DoDress-T53B (D-BERT)	40.5
DoDress-T53B (GPL)	43.2
Proposé : T53B, Négatifs durs BM25	
DoDress-BM25 (D-BERT)	41.6
DoDress-BM25 (GPL)	43.3
Proposé : T53B, Négatifs durs SimANS	
DoDress-T53B (D-BERT)	**43.6**
DoDress-T53B (GPL)	**45.5**

TABLE 5 – Résultat d'adaptation de domaine de BioASQ.

modèle	nDCG@10 (%)
Modèles sans adaptation	
D-BERT	53.6
BM25 (Anserini)	73.0
Re-Ranking avec des Cross-Encoders (limite supérieure)	
BM25 + CE	72.8
BM25 + T53B	76.1
Méthodes précédentes d'adaptation de domaine	
UDALM	33.1
MoDIR (ANCE)	47.9
Pré-entraînement basé : Cible → D-BERT	
SimCSE	53.2
ICT	55.3
TSDAE	55.5
Basé sur la génération (SOTA précédent)	
QGen	56.5
GPL	62.8
TSDAE + GPL	61.6
Proposée : T53B, Négatifs Aléatoires Globaux	
DoDress-T53B (D-BERT)	52.9
DoDress-T53B (GPL)	62.0
Proposé : T53B, Négatifs durs BM25	
DoDress-BM25 (D-BERT)	58.6
DoDress-BM25 (GPL)	64.7
Proposé : T53B, Négatifs durs SimANS	
DoDress-T53B (D-BERT)	**60.6**
DoDress-T53B (GPL)	**65.3**

cible.

Nous avons également étudié l'importance du choix de la stratégie d'échantillonnage de documents non pertinents. Les meilleurs résultats sont obtenus en utilisant SimANS, une stratégie récente d'échantillonnage de documents non pertinents à partir des listes de documents obtenues par le modèle dense que l'on cherche à déployer dans le domaine cible.

Remerciements

Ce travail a été partiellement financé par MIAI@Grenoble Alpes (ANR-19-P3IA-0003) et la bourse du Chinese Scholarship Council (CSC) numéro 201906960018.

Références

[1] BLANCHARD G., DESHMUKH A. A., DOGAN Ü., LEE G. & SCOTT C. (2021). Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, **22**(1), 46–100.

[2] BURGES C. J. (2010). From ranknet to lambdarank to lambdamart : An overview. *MSR-Tech Report*.

[3] CARLUCCI F. M., D'INNOCENTE A., BUCCI S., CAPUTO B. & TOMMASI T. (2019). Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 2229–2238.

[4] 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.

[5] DU Y., XU J., XIONG H., QIU Q., ZHEN X., SNOEK C. G. & SHAO L. (2020). Learning to learn with variational information bottleneck for domain generalization. In *European Conference on Computer Vision*, p. 200–216 : Springer.

[6] D'INNOCENTE A. & CAPUTO B. (2018). Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, p. 187–198 : Springer.

[7] GANIN Y., USTINOVA E., AJAKAN H., GERMAIN P., LAROCHELLE H., LAVIOLETTE F., MARCHAND M. & LEMPITSKY V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, **17**(1), 2096–2030.

[8] GAO L. & CALLAN J. (2021). Condenser : a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 981–993.

[9] GAO L. & CALLAN J. (2022). Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2843–2853.

[10] GAO T., YAO X. & CHEN D. (2021). Simcse : Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 6894–6910.

[11] GUO J., FAN Y., PANG L., YANG L., AI Q., ZAMANI H., WU C., CROFT W. B. & CHENG X. (2020). A deep look into neural ranking models for information retrieval. *Information Processing & Management*, **57**(6), 102067.

[12] HOFSTÄTTER S., ALTHAMMER S., SCHRÖDER M., SERTKAN M. & HANBURY A. (2020). Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv :2010.02666*.

[13] JEON S., HONG K., LEE P., LEE J. & BYUN H. (2021). Feature stylization and domain-aware contrastive learning for domain generalization. In *Proceedings of the 29th ACM International Conference on Multimedia*, p. 22–31.

[14] KAROUZOS C., PARASKEVOPOULOS G. & POTAMIANOS A. (2021). Udalm : Unsupervised domain adaptation through language modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 2579–2590.

[15] KARPUKHIN V., OGUZ B., MIN S., LEWIS P., WU L., EDUNOV S., CHEN D. & YIH W.-T. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6769–6781.

[16] KUN ZHOU, YEYUN GONG X. L. W. X. Z. Y. S. A. D. J. L. R. M. J.-R. W. N. D. & CHEN W. (2022). Simans : Simple ambiguous negatives sampling for dense text retrieval.

[Laignelet & Rioult] LAIGNELET M. & RIOULT F. Repérer automatiquement les segments obsolescents à l'aide d'indices sémantiques et discursifs.

[Langlais & Patry] LANGLAIS P. & PATRY A. Enrichissement d'un lexique bilingue par analogie. p. 101–110.

[19] LEE K., CHANG M.-W. & TOUTANOVA K. (2019). Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 6086–6096.

[20] LI D., YANG Y., SONG Y.-Z. & HOSPEDALES T. (2018). Learning to generalize : Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[21] LI D., YANG Y., SONG Y.-Z. & HOSPEDALES T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, p. 5542–5550.

[22] LI M. & GAUSSIER E. (2022). Bert-based dense intra-ranking and contextualized late interaction via multi-task learning for long document retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 2347–2352.

[23] LIANG D., XU P., SHAKERI S., SANTOS C. N. D., NALLAPATI R., HUANG Z. & XIANG B. (2020). Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv :2009.10270*.

[24] LIU C., SUN X., WANG J., TANG H., LI T., QIN T., CHEN W. & LIU T.-Y. (2021). Learning causal semantic representation for out-of-distribution prediction. *Advances in Neural Information Processing Systems*, **34**, 6155–6170.

[25] LOSHCHILOV I. & HUTTER F. (2017). SGDR : Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

[26] MA J., KOROTKOV I., YANG Y., HALL K. & MCDONALD R. (2021). Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 1075–1088.

[27] 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 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.

[28] MANCINI M., BULO S. R., CAPUTO B. & RICCI E. (2018). Best sources forward : domain generalization through source-specific nets. In *2018 25th IEEE international conference on image processing (ICIP)*, p. 1353–1357 : IEEE.

[29] MICIKEVICIUS P., NARANG S., ALBEN J., DIAMOS G., ELSEN E., GARCIA D., GINSBURG B., HOUSTON M., KUCHAIEV O., VENKATESH G. *et al.* (2018). Mixed precision training. In *International Conference on Learning Representations*.

[30] MOKRII I., BOYTSOV L. & BRASLAVSKI P. (2021). A systematic evaluation of transfer learning and pseudo-labeling with bert-based ranking models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 2081–2085.

[31] MOTIIAN S., PICCIRILLI M., ADJEROH D. A. & DORETTO G. (2017). Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, p. 5715–5725.

[32] NAM H., LEE H., PARK J., YOON W. & YOO D. (2021). Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 8690–8699.

[33] NGUYEN T., ROSENBERG M., SONG X., GAO J., TIWARY S., MAJUMDER R. & DENG L. (2016). MS MARCO : A human generated machine reading comprehension dataset. In T. R. BESOLD, A. BORDES, A. S. D'AVILA GARCEZ & G. WAYNE, Éds., *Proceedings of the Workshop on Cognitive Computation : Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 de *CEUR Workshop Proceedings* : CEUR-WS.org.

[34] NOGUEIRA R., JIANG Z., PRADEEP R. & LIN J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 708–718.

[35] PRAKASH A., BOOCHOON S., BROPHY M., ACUNA D., CAMERACCI E., STATE G., SHAPIRA O. & BIRCHFIELD S. (2019). Structured domain randomization : Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, p. 7249–7255 : IEEE.

[36] QIAO F., ZHAO L. & PENG X. (2020). Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 12556–12565.

[37] RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W., LIU P. J. *et al.* (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, **21**(140), 1–67.

[38] RAHMAN M. M., FOOKES C., BAKTASHMOTLAGH M. & SRIDHARAN S. (2019). Multi-component image translation for deep domain generalization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, p. 579–588 : IEEE.

[39] 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.

[40] ROBERTSON S. E. & ZARAGOZA H. (2009). The probabilistic relevance framework : BM25 and beyond. *Found. Trends Inf. Retr.*, **3**(4), 333–389.

[41] SANH V., DEBUT L., CHAUMOND J. & WOLF T. (2019). Distilbert, a distilled version of BERT : smaller, faster, cheaper and lighter. *arXiv preprint arXiv :1910.01108*.

[Seretan & Wehrli] SERETAN V. & WEHRLI E. Collocation translation based on sentence alignment and parsing. p. 401–410.

[43] SHANKAR S., PIRATLA V., CHAKRABARTI S., CHAUDHURI S., JYOTHI P. & SARAWAGI S. (2018). Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*.

[44] THAKUR N., REIMERS N., RÜCKLÉ A., SRIVASTAVA A. & GUREVYCH I. (2021). BEIR : A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[45] TOBIN J., FONG R., RAY A., SCHNEIDER J., ZAREMBA W. & ABBEEL P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, p. 23–30 : IEEE.

[46] TSATSARONIS G., BALIKAS G., MALAKASIOTIS P., PARTALAS I., ZSCHUNKE M., ALVERS M. R., WEISSENBORN D., KRITHARA A., PETRIDIS S., POLYCHRONOPOULOS D. *et al.* (2015). An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, **16**(1), 138.

[47] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

[48] VOLPI R., NAMKOONG H., SENER O., DUCHI J. C., MURINO V. & SAVARESE S. (2018). Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, **31**.

[49] VOORHEES E. (2005). : Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD.

[50] WANG J., LAN C., LIU C., OUYANG Y., QIN T., LU W., CHEN Y., ZENG W. & YU P. (2022a). Generalizing to unseen domains : A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

[51] WANG K., REIMERS N. & GUREVYCH I. (2021). Tsdae : Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics : EMNLP 2021*, p. 671–688.

[52] WANG K., THAKUR N., REIMERS N. & GUREVYCH I. (2022b). GPL : Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human*

Language Technologies, p. 2345–2360, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.168.

[53] WANG M. & DENG W. (2018). Deep visual domain adaptation : a survey. *Neurocomputing*.

[54] XIN J., XIONG C., SRINIVASAN A., SHARMA A., JOSE D. & BENNETT P. (2022). Zero-shot dense retrieval with momentum adversarial domain invariant representations. In *Findings of the Association for Computational Linguistics : ACL 2022*, p. 4008–4020.

[55] XIONG L., XIONG C., LI Y., TANG K.-F., LIU J., BENNETT P. N., AHMED J. & OVERWIJK A. (2020). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

[56] YANG P., FANG H. & LIN J. (2018). Anserini : Reproducible ranking baselines using lucene. *Journal of Data and Information Quality (JDIQ)*, **10**(4), 1–20.

[57] ZHANG H., CISSE M., DAUPHIN Y. N. & LOPEZ-PAZ D. (2018). mixup : Beyond empirical risk minimization. In *International Conference on Learning Representations*.

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

Extraction d'entités nommées à partir de descriptions d'espèces

Maya Sahraoui[1, 2] Vincent Guigue[3] Regine Vignes-Lebbe[2] Marc Pignal[2]
(1) ISIR, Sorbonne Université, Paris , France
(2) MNHN, Sorbonne Université, Paris , France
(3) AgroParisTech, Paris-Saclay, Paris, France
sahraoui@isir.upmc.fr, vincent.guigue@isir.upmc.fr,
regine.vignes*ı*ebbe@$sorbonne - universite.fr$, $marc.pignal@mnhn.fr$

RÉSUMÉ

Les descriptions d'espèces contiennent des informations importantes sur les caractéristiques morphologiques des espèces, mais l'extraction de connaissances structurées à partir de ces descriptions est souvent chronophage. Nous proposons un modèle texte-graphe adapté aux descriptions d'espèces en utilisant la reconnaissance d'entités nommées (NER) faiblement supervisée. Après avoir extrait les entités nommées, nous reconstruisons les triplets en utilisant des règles de dépendance pour créer le graphe. Notre méthode permet de comparer différentes espèces sur la base de caractères morphologiques et de relier différentes sources de données. Les résultats de notre étude se concentrent sur notre modèle NER et démontrent qu'il est plus performant que les modèles de référence et qu'il constitue un outil précieux pour la communauté de l'écologie et de la biodiversité.

ABSTRACT

Named Entity Recognition for species descriptions.

Species descriptions contain important information about the morphological characteristics of species, but extracting structured knowledge from these descriptions is often time consuming. We propose a text-graph model adapted to species descriptions using weakly supervised named entity recognition (NER). After extracting the named entities, we reconstruct the triplets using dependency rules to create the graph. Our method allows us to compare different species based on morphological characters and link different data sources. The results of our study focus on our NER model and demonstrate that it outperforms benchmark models and is a valuable tool for the ecology and biodiversity community.

MOTS-CLÉS : Extraction d'entités nommées$_1$, Supervision distante$_2$, Bases de connaissance$_3$..

KEYWORDS: Named entity recognition$_1$, Distant supervision$_2$, Knowledge base$_3$..

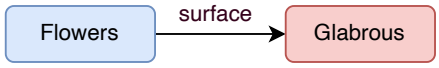
Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA), pages 111–126.

1 Introduction

Les descriptions d'espèces sont une source d'information cruciale pour les études sur la biodiversité et l'identification des espèces. Elles fournissent des informations détaillées sur les caractéristiques morpho-anatomiques d'une espèce donnée, qui sont essentielles pour la distinguer des autres espèces. Ces descriptions sont généralement rédigées sous forme de texte dans les revues scientifiques, les livres ou les bases de données. Pour faciliter l'étude et la comparaison de différentes espèces, les descriptions d'espèces sont souvent représentées sous la forme de graphes de connaissances qui sont utilisés pour identifier les relations entre différentes espèces et modéliser l'évolution des espèces. Cependant, l'extraction de ces informations à partir des descriptions d'espèces peut être une tâche longue et laborieuse, généralement réalisée par des experts (Saucède *et al.*, 2021). L'automatisation de ce processus est donc un enjeu pour les chercheurs de cette communauté.

Les modèles d'apprentissage profond ont montré des performances remarquables dans les tâches d'extraction de connaissances (Miwa & Bansal, 2016). Cependant, ces modèles sont lourds et nécessitent une quantité importante de données annotées dans le domaine visé pour atteindre une grande précision, ce qui est très couteux. Nous nous focalisons ainsi sur les approches d'extraction d'entités nommées (NER) en faisant l'hypothèse que le domaine particulier des documents visés nous permettra de créer les étiquettes spécifiques pour les sources –*organes*– et les cibles –*types spécifiques d'attributs*–. Le fait que les attributs soient identifiés avec leur nature permet la reconstruction du triplet comme dans l'exemple :

<ORGAN> Flowers </ORGAN> 4-merous. Calyx aestivation valvate, campanulate, 2-3.6mm long, abaxially <DESC-SURFACE> glabrous </DESC-SURFACE>



La plupart des corpus annotés et des modèles pré-entraînés en NER sont conçus pour des données générales (e.g. wikipedia). De plus, le transfert entre domaines est difficile (Taillé *et al.*, 2021). A l'exception des documents financiers ou biomédicaux (Magge *et al.*, 2018), peu de domaines spécifiques sont abordables en exploitant seulement les ressources académiques disponibles. L'enjeu de cet article est donc très appliqué : il s'agit de proposer une méthodologie globale pour l'extraction d'information dans le domaine de la description d'espèces.

Notre approche repose sur 3 étapes : (1) Segmenter le texte par description d'organe à l'aide de règles ; (2) Identifier les organes et les descripteurs par extraction d'entités nommées ; (3) Reconstruire les triplets à l'aide de règles. Cet article décrit principalement le processus de l'étape (2) qui se décompose lui-même en différentes étapes : (a) Récupérer un glossaire (mots-clés et classes) auprès d'experts du domaine ; (b) Exploiter les termes du glossaire pour annoter le corpus de manière distante ; (c) Entrainer un premier modèle NER ; (d) Améliorer le modèle (teacher-student self-training). La contribution applicative de cet article est donc d'étudier un cas d'usage spécifique de l'extraction d'entités nommées en supervision distante avec les outils de l'état de l'art.

Après une rapide bibliographie sur la gestion et l'extraction de connaissance, en particulier sur les approches de NER apprise en supervision distante (section 2), nous décrirons en section 3 le processus d'annotation et le modèle utilisé pour l'extraction. La section 4 décrit les résultats quantitatifs et qualitatifs obtenus lors de la campagne d'expériences.

2 Travaux connexes

La représentation des connaissances est un enjeu important pour l'étude des espèces en biologie. Les systèmes actuels reposent principalement sur des experts (Vignes-Lebbe *et al.*, 2017), ce qui est très intéressant pour des études ciblées mais qui ne permet pas le traitement en masse des documents disponibles. L'enjeu consiste à trouver les bons outils dans la littérature pour faire face à ce défi. Les modèles d'apprentissage profond ont montré des performances remarquables dans les tâches d'extraction de connaissances mais nécessitent en général un large corpus de données annotés dans le domaine pour l'apprentissage (Miwa & Bansal, 2016; Taillé *et al.*, 2021).

En réduisant la tâche d'extraction de connaissances à de la reconnaissance d'entités nommées (NER), comme illustré en introduction, les architectures sont plus simples. Les architectures de NER ont beaucoup évoluées ces dernières années : les chaines de Markov cachées (HMM) puis les champs aléatoires conditionnels (CRF) ont permis des avancées significatives dans la modélisation et l'analyse des séquences de mots au début des années 2000 (Malouf, 2002; McCallum & Li, 2003). Les approches neuronales (convolutionnelles puis récurrentes) ont ensuite largement contribué à l'amélioration des performances (Collobert *et al.*, 2011) mais c'est la combinaison de différentes représentations des mots (Mikolov *et al.*, 2013) à l'entrée de ces architectures neuronales qui a engendré les gains de performances les plus importants (Tai *et al.*, 2015; Lample *et al.*, 2016). L'avènement de l'architecture Transformer (Vaswani *et al.*, 2017) et des modèles dérivés (Devlin *et al.*, 2019) ont entretenu la dynamique d'amélioration des perfromances à la fin des années 2010. Malgré cette dynamique remarque, il reste nécessaire d'avoir des données étiquetées du domaine du fait des capacités de transfert limitées des approches NER (Taillé *et al.*, 2020).

Cette limite sur le transfert explique l'émergence d'approches (et de corpus) spécifiques pour différentes langues (Souza *et al.*, 2019; Jia *et al.*, 2020) ou différents type de données comme les données biomédicales (Cho & Lee, 2019), légales (Leitner *et al.*, 2019) ou financières (Lee *et al.*, 2022). Cela montre aussi à quel point l'extraction d'entités nommées reste encore aujourd'hui une tâche particulièrement délicate en TAL malgré les avancées récentes.

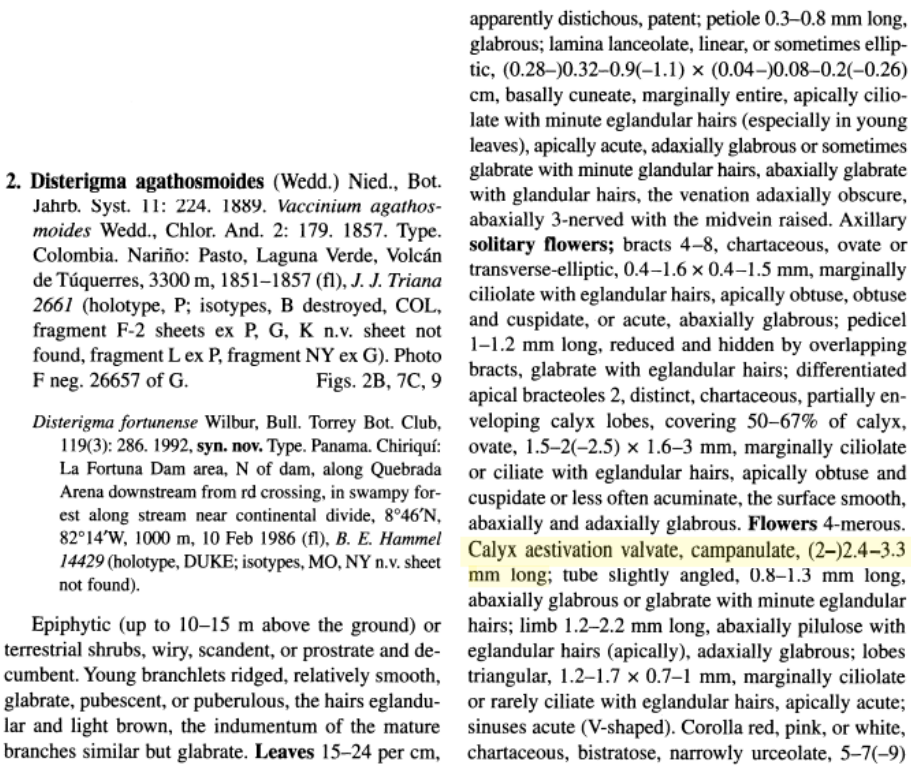
Le coût d'un corpus annoté en NER est conséquent, ce qui nous a poussé à étudier les possibilités de supervision distante (Wang *et al.*, 2020, 2021). A partir d'une liste de termes catégorisés (ie un glossaire réalisé par un expert), nous utilisons des expressions régulières pour annoter un corpus d'apprentissage. La supervision distante est par essence imparfaite : certaines annotations sont manquantes, d'autres ambigues voire erronées. Ce type de corpus implique donc des approches robustes qui seront en mesure de moins pénaliser certaines erreurs pour améliorer la généralisation. L'auto-apprentissage est actuellement l'approche la plus efficace pour éviter le sur-apprentissage d'une part et forcer le modèle à découvrir de nouveaux termes afin d'améliorer le rappel d'autre part. L'intégration séquentielle d'étiquettes prédites avec une forte confiance dans l'ensemble d'apprentissage pose cependant un grand risque de dérive du modèle qui peut être contenu en recourant à une architecture teacher-student (Liang *et al.*, 2020). Il est également pertinent de ré-entrainer le modèle de langue, à découvrir des mots masqués dans le contexte du domaine cible (Meng *et al.*, 2021). Nous allons exploiter ces deux stratégies dans le modèle proposé dans la section suivante.

3 Travaux et méthodes

Nous décrivons d'abord les données brutes et leur préparation, c'est-à-dire le processus d'annotation distante pour les entités. Nous détaillerons ensuite l'architecture envisagée pour la tâche de NER et la mise en œuvre de l'auto-apprentissage.

3.1 Création du jeu de données

Les corpus de faune et de flore contiennent des descriptions textuelles détaillées des espèces, y compris les caractéristiques morphologiques de divers groupes taxonomiques tels que les espèces, les genres et les familles. Notre travail se concentrera principalement sur les flores et en particulier le corpus *Flora Neotropica*, qui comprend des clés et des descriptions de différentes espèces, ainsi que des informations géographiques, cf Figure 1.



2. Disterigma agathosmoides (Wedd.) Nied., Bot. Jahrb. Syst. 11: 224. 1889. *Vaccinium agathosmoides* Wedd., Chlor. And. 2: 179. 1857. Type. Colombia. Nariño: Pasto, Laguna Verde, Volcán de Túquerres, 3300 m, 1851–1857 (fl), *J. J. Triana 2661* (holotype, P; isotypes, B destroyed, COL, fragment F-2 sheets ex P, G, K n.v. sheet not found, fragment L ex P, fragment NY ex G). Photo F neg. 26657 of G. Figs. 2B, 7C, 9

Disterigma fortunense Wilbur, Bull. Torrey Bot. Club, 119(3): 286. 1992, **syn. nov.** Type. Panama. Chiriquí: La Fortuna Dam area, N of dam, along Quebrada Arena downstream from rd crossing, in swampy forest along stream near continental divide, 8°46′N, 82°14′W, 1000 m, 10 Feb 1986 (fl), *B. E. Hammel 14429* (holotype, DUKE; isotypes, MO, NY n.v. sheet not found).

Epiphytic (up to 10–15 m above the ground) or terrestrial shrubs, wiry, scandent, or prostrate and decumbent. Young branchlets ridged, relatively smooth, glabrate, pubescent, or puberulous, the hairs eglandular and light brown, the indumentum of the mature branches similar but glabrate. **Leaves** 15–24 per cm, apparently distichous, patent; petiole 0.3–0.8 mm long, glabrous; lamina lanceolate, linear, or sometimes elliptic, (0.28–)0.32–0.9(–1.1) × (0.04–)0.08–0.2(–0.26) cm, basally cuneate, marginally entire, apically ciliolate with minute eglandular hairs (especially in young leaves), apically acute, adaxially glabrous or sometimes glabrate with minute glandular hairs, abaxially glabrate with glandular hairs, the venation adaxially obscure, abaxially 3-nerved with the midvein raised. Axillary **solitary flowers;** bracts 4–8, chartaceous, ovate or transverse-elliptic, 0.4–1.6 × 0.4–1.5 mm, marginally ciliolate with eglandular hairs, apically obtuse, obtuse and cuspidate, or acute, abaxially glabrous; pedicel 1–1.2 mm long, reduced and hidden by overlapping bracts, glabrate with eglandular hairs; differentiated apical bracteoles 2, distinct, chartaceous, partially enveloping calyx lobes, covering 50–67% of calyx, ovate, 1.5–2(–2.5) × 1.6–3 mm, marginally ciliolate or ciliate with eglandular hairs, apically obtuse and cuspidate or less often acuminate, the surface smooth, abaxially and adaxially glabrous. **Flowers** 4-merous. Calyx aestivation valvate, campanulate, (2–)2.4–3.3 mm long; tube slightly angled, 0.8–1.3 mm long, abaxially glabrous or glabrate with minute eglandular hairs; limb 1.2–2.2 mm long, abaxially pilulose with eglandular hairs (apically), adaxially glabrous; lobes triangular, 1.2–1.7 × 0.7–1 mm, marginally ciliolate or rarely ciliate with eglandular hairs, apically acute; sinuses acute (V-shaped). Corolla red, pink, or white, chartaceous, bistratose, narrowly urceolate, 5–7(–9)

FIGURE 1 – Echantillon du corpus *Flora Neotropica* contenant la description de l'espèce *Disterigma agathosmoides*

Les descriptions morphologiques des espèces dans les flores suivent un format semi-structuré qui passe progressivement d'une description d'un organe à ses sous-organes apparentés. Chaque organe ou sous-organe est décrit à l'aide de plusieurs descripteurs, notamment la couleur, la forme, la position et la disposition. Après avoir analysé les descriptions morphologiques, nous avons identifié trois schémas distincts dans la syntaxe des descriptions. La ponctuation (point et point-virgule) joue un rôle prépondérant dans la segmentation des descriptions disponibles :

Schéma 1 : Organ 1 description. Organ 2 description.

Schéma 2 : Organ 1 description ; Organ 2 description.
Schéma 3 : Main organ 1 description ; sub organ 1 ; sub organ 2 ; sub organ 3. Main organ 2 description ; sub organ 1 ; sub organ 2 ; sub organ 3.

Nous faisons l'hypothèse qu'en divisant les descriptions selon ces schémas, chaque segment de texte est centrée sur un organe ou un sous-organe. Après la segmentation, notre système va extraire l'organe en question (parfois malheureusement non unique) et les descripteurs. L'astuce, mentionnée en introduction, consiste à typer les entitées descripteurs à la fois avec une catégorie (couleur, forme, position,...) et à considérer ces entités comme des valeurs (vert, acuminé, alterné,...). Ainsi, les triplets modélisant les connaissances seront formés, dans chaque segment, de la manière suivante : le sujet sera l'organe principal, le prédicat correspondra à la catégorie et l'objet sera la valeur du descripteur extrait.

Soit G un glossaire contenant des mots liés à des descriptions morphologiques. Chaque mot w_j du glossaire est associé à une étiquette spécifique y_j.

$$G = \{(w_1, y_1), \dots (w_j, y_j), \dots, (w_N, y_N)\}, \qquad (w_j, y_j) \in \mathcal{W} \times \mathcal{Y} \tag{1}$$

Par soucis de simplicité, l'expert n'a dans un premier temps lister que des mots simples : nous n'abordons donc pas ici la problématique des entitées composées de plusieurs mots. L'ensemble $\mathcal{Y} = \mathcal{Y}_0 \cup \mathcal{Y}_1$ est constitué de deux sous-ensembles d'étiquettes correspondant respectivement aux organes et aux descripteurs :

$$\mathcal{Y}_0 = \{Flower, Fruit, Habit, Leaf, Part\text{-}of, Stem\text{-}root\} \tag{2}$$

$$\mathcal{Y}_1 = \{Color, Disposition, Form, Position, Surface\text{-}texture\} \tag{3}$$

Attention, dans toute la suite, les y désigneront en réalité des vecteurs de scores sur les C classes ($y \in \mathbb{R}^C$). Dans la version initiale, y prend la forme d'un *one-hot* sur la classe visée. Dans la suite de l'article, pour faire simple, nous parlerons de classe y alors que formellement, la classe serait plutôt $\text{argmax}(y)$. Cette subtilité est importante pour pouvoir introduire les mécanismes d'auto-supervision dans la suite.

Processus d'annotation distante. Après application des schémas sus-mentionnés, nous considérons le corpus comme un ensemble de phrases $S = \{s_0, s_1, ..., s_M\}$, chaque phrase $s_m = \{w_0, w_1, ..., w_{N_m}\}$ étant un ensemble de mots w_n.

Les mots w du corpus S et du glossaire G sont lemmatisés puis comparés : chaque appariement permet d'annoter un mot w_n avec l'étiquette y du glossaire. Les mots non reconnus sont affectés à la classe O. Nous obtenons ainsi un ensemble de séquences d'étiquettes $L = \{\ell_1, \dots, \ell_M\}$, $\ell_m = \{y_1, \dots, y_{N_m}\}$ alignées avec le corpus de phrases S.

Les statistiques du glossaire et de sa projection sur le corpus sont données dans le tableau 1.

TABLE 1 – Statistiques du jeu de données : classes considérées, nombres de mots distincts dans claque classe et nombre d'occurences dans le corpus.

Ensemble	Classe	Nombre d'occurences	Nombre de mots
$\mathcal{Y}_0$	Flower	22890	23
	Fruit	4968	10
	Habit	1920	3
	Leaf	4364	5
	Part-of	23849	25
	Stem-root	3296	7
$\mathcal{Y}_1$	Color	18342	15
	Disposition	8405	21
	Form	24816	64
	Position	10936	13
	Surface-texture	18325	23

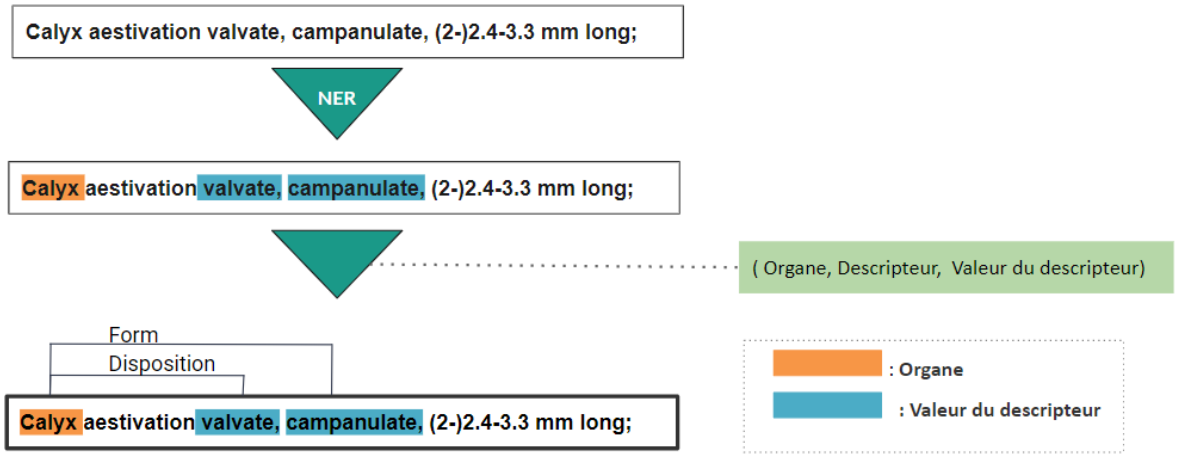


FIGURE 2 – Processus d'extraction de triplets à partir d'une phrase centrée sur un organe

3.2 Méthode d'extraction d'entitées nommées proposée

L'extraction d'entités nommées est une tâche difficile, ainsi, le développement d'une telle approche à partir de données aux annotations **bruitées** et **incomplètes** est un défi scientifique. La question même de l'évaluation dans un tel cadre est non triviale, nous y reviendrons plus tard. Pour résoudre ce problème, nous avons adopté l'auto-apprentissage, une technique qui intègre itérativement les prédictions les plus fiables du modèle dans la vérité terrain pour améliorer progressivement la couverture du modèle.

Le second défi auquel nous avons été confrontés réside dans la spécificité du vocabulaire et des structures de phrases associée à un domaine aussi pointu. Comme le montre la Figure 1, la distribution du vocabulaire et l'organisation des séquences de mots diffèrent trop du langage naturel usuel pour bénéficier de l'apport des modèles de langue pré-entrainés. Nous avons étudié deux stratégies pour palier ce problème : (1) l'utilisation d'un modèle de langue pré-entrainé, en misant sur la robustesse de celui-ci et (2) l'affinage du modèle de langues pré-entrainé en refaisant des prédictions de mots masqués sur les documents issus des flores.

Architecture. Le modèle de reconnaissance des entités nommées repose sur un encodeur BERT pré-entraîné avec une couche entièrement connectée pour la classification par mots-clés. Nous désignons ce modèle par f_θ, où θ représente l'ensemble des paramètres (modèle de langue et couche de classification). Pour chaque séquence s_m, $f_\theta(s_m) \in \mathbb{R}^{C \times N_m}$ est une estimation de $p(Y_j = c|s_m)$ pour tous les mots w_j de s et pour les C classes considérées. Nous noterons $f_{\theta,j,c}(w_j)$ la prédiction associée au mot j et à la classe c. L'entrainement du modèle est effectué sur des paires de séquences alignées (s_m, ℓ_m). La fonction de coût est classiquement une entropie croisée :

$$\mathcal{L} = - \sum_{(s,\ell) \in S,L} \sum_{(w_j,y_j) \in (s,\ell)} \sum_{c=0}^{C} y_{j,c} \log \frac{\exp(f_{\theta,j,c}(s))}{\sum_{c'=1}^{C} \exp(f_{\theta,j,c'}(s))} \tag{4}$$

Note : rappelons comme indiqué en section 3.1, que les y sont en fait des vecteurs de score sur les classes. $y_{j,c}$ désigne ainsi le score du mot j pour la classe c. Dans cette première partie, $y_{j,c} = 1$ pour la classe annotée de w_j et 0 pour toutes les autres classes.

Ré-entrainement de toutes les couches du modèle. Au cours du processus d'apprentissage, nous mettons à jour toutes les couches du modèle de bout en bout : les couches de l'encodeur ne sont pas figées. Cette approche améliore la qualité des résultats et l'efficacité d'apprentissage en modifiant les dernieres couches de l'encodeur pour les adapter à la tâche de classification.

Pré-entrainement du modèle de langue. Le pré-entraînement de l'encodeur BERT sur la tâche non supervisée de prédiction de mots masqués peut améliorer de manière significative la qualité des représentations apprises lorsque le domaine textuel est très particulier, comme c'est le cas dans notre application. Nous avons donc mis en place une précédure par entraîner un classifieur de mots $g_{\theta'}$ sur un corpus large et diversifié de textes biologiques, en particulier sur des données de descriptions d'espèces (différentes de celles du corpus utilisé pour le NER). Une fois quelques itérations effectuées, les paramètres de la couche de classification sont éliminés et les paramètres du modèle de langue θ' sont transférés en initialisation du modèle NER qui devient $f_{\theta'}$. Cette procédure permet au modèle NER de bénéficier du pré-entraînement.

3.3 Apprentissage auto-supervisé

La procédure d'auto-supervision est une approche itérative. Nous utilisons la stratégie décrite dans (Liang *et al.*, 2020).

Initialisation du *teacher*. Définissons d'abord un modèle de NER de référence f_θ^T entrainé sur notre jeu de données : ce modèle est appelé *teacher* (T). A l'itération 1, $f_\theta^{(T)}$ génère une liste de prédictions $\hat{Y} = \text{Softmax}(f_\theta^{(T)}(S)))$ associées aux phrases du corpus. Les prédictions dont le score de confiance dépassent un seuil fixé γ sont utilisées pour corriger les labels Y. Nous notons ce nouvel étiquetage $Y^{(1)}$.

$$\forall j,\ y_j^{(1)} = \begin{cases} y_j & \text{si argmax}(\hat{y}_j) = \text{argmax}(y_j) \text{ [bonne classification]} \\ \hat{y}_j & \text{si } \max(\hat{y}_j) > \gamma \text{ et argmax}(y_j) = 0 \\ y_j & \text{sinon} \end{cases} \tag{5}$$

Initialisation du *student*. Le modèle *student* $f_{\theta'}^{(S)}$ est appris sur ce jeu de données *corrigées*. En reprenant l'équation (4), l'intérêt de la procédure est plus clair : le fait de considérer la distribution

des scores sur y_j permet d'éviter des changements trop brusques dans l'étiquetage et de stabiliser l'évolution des modèles [1].

Itérations du *student*. Le modèle *student* $f_{\theta'}^{(S)}$ prédit successivement de nouvelles étiquettes $Y^{(t)}$, selon la procédure décrite en équation (5), puis met à jour ses poids θ' en exploitant $Y^{(t)}$.

Cette procédure présente un risque évident de dérive, le modèle se confortant progressivement dans des propositions fausses émanant de lui-même. Le modèle BOND (Liang *et al.*, 2020) propose de réinitialiser régulièrement les poids θ' du modèle à θ (les poids du *teacher*) : les étiquettes évoluent continuellement mais le risque de dérive est limité par un retour périodique au modèle d'origine.

Le nombre d'itérations $N_{self\text{-}training}$ et la période de retour à l'origine N_{reinit} sont des hyperparamètres très sensibles pour éviter la divergence.

3.4 Module de reconstruction des triplets

Sur la base de nos hypothèses concernant les schémas trouvés dans les descriptions d'espèces dans les corpus Flora neotropica (cf Section 3), nous avons conçu un module de reconstruction de triplets qui exploite les sorties du modèle d'extraction d'entités nommées. Nous supposons que, grâce à notre méthode d'échantillonnage, chaque phrase est centrée sur un organe ou un sous-organe particulier : il suffit alors de rattacher les descripteurs de la même phrase à cet organe.

$$\begin{aligned}&\{(w_{org}, y_{desc_0}, w_{desc_0}),\\ &(w_{org}, y_{desc_1}, w_{desc_1}), ...,\\ &(w_{org}, y_{desc_{N'}}, w_{desc_{N'}})\}\end{aligned} \tag{6}$$

Dans les rares cas où plusieurs organes sont détectés dans la même phrase, la meilleure solution consiste simplement à retenir la première détection dans l'ordre de la phrase.

4 Expériences

Dans cette section, nous présentons la campagne d'expériences concernant l'extraction des entités nommées. L'évaluation des triplets extraits est une tâche très importante pour les experts mais coûteuse en interventions humaines. Par conséquent, elle ne sera pas traitée dans cette étude.

Nous envisageons dans cette section plusieurs variantes de notre modèle pour la reconnaissance d'entités nommées (NER) sur les descriptions d'espèces.

Baseline : L'architecture de référence pour la reconnaissance des entités nommées utilisée est un modèle BERT-base pré-entraîné sur lequel est superposée une couche de classification avec une adaptation complète du modèle. Le pas d'apprentissage a été fixé à 10^{-6} et le nombre d'itérations nécessaire à la convergence du modèle est de 26. Le critère de convergence est calculé sur le score f1 en validation.

Baseline with pre-trained language model : Pour cette variante, nous pré-entraînons le modèle de langue de BERT-base sur l'ensemble des données de descriptions d'espèces en utilisant

1. Il s'agit d'une des variantes étudiées dans (Liang *et al.*, 2020), elle s'est révélée la plus performante sur nos données.

la tâche de prédiction de mots masqués. Nous ré-entrainons ensuite le modèle pour l'extraction d'entités nommées. Le pas d'apprentissage a été fixé à 10^{-6} et le nombre d'itérations nécessaire à la convergence du modèle est de 26 (score f1 maximal en validation).

Baseline with self training : Le modèle initial *teacher* est initialisé avec les paramètres du modèle **Baseline**. Le processus d'auto-apprentissage décrit dans la section 3.3 est ensuite appliqué pour entraîner le modèle. Le pas d'apprentissage a été fixé à 10^{-6} et le nombre d'itérations nécessaire à la convergence du modèle est de 4, le seuil de confiance γ a été fixé à 0.9 et le modèle *student* est réinitialisé 2 fois par epoch.

Baseline with self-training and language model pre-training : Le modèle initial *teacher* est cette fois initialisé avec les paramètres du modèle **Baseline with pre-trained language model** avant d'appliquer le processus d'auto-apprentissage. Le pas d'apprentissage a été fixé à 10^{-6} et les performances du modèle en validation ont commencé à chuter au bout de 3 itérations, le seuil de confiange γ a été fixé à 0.9 et le modèle *student* est réinitialisé 2 fois par epoch.

Afin d'évaluer l'efficacité des méthodes , nous calculons le score F1, la précision et le rappel pour chaque variante. Pour une classe de données c, nous calculons classiquement :

$$F1_c = 2 \cdot \frac{\text{Precision} \cdot \text{Rappel}}{\text{Precision} + \text{Rappel}} \qquad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

en utilisant les abbréviations anglaise ; TP (vrai positif), FP (faux positif), FN (faux négatif). Nous calculons ensuite le score F1 micro pondéré par le poids des classes :

$$F1_{weighted} = \frac{\sum_{c=1}^{N_{cl}} N_{occ_c} \times F1_c}{\sum_{c=1}^{N_{cl}} N_{occ_c}} \tag{8}$$

Où N_{cl} représente le nombre total de classes et N_{occ_c} représente le nombre d'occurences de la classe c dans le jeu de données de test.

Classe 0 *vs* organes et descripteurs. Comme c'est toujours le cas en détection d'entités nommées, la classe 0, ultra-majoritaire, est bien entendu exclue du calcul $F1_{weighted}$.

Détection *vs* classification. Chaque métrique peut être calculée à deux niveaux : la **détection** (niveau le plus lâche) consiste à quantifier les termes qui ont été détectés, indépendemment de la classe affectée. La **classification** représente dans cet article la métrique la plus stricte, intégrant à la fois la détection et la bonne classification des termes.

Ces mesures fournissent une évaluation de la capacité du modèle à identifier et à extraire avec précision des entités à partir de données textuelles, ainsi que sa sensibilité aux entités nouvelles ou inédites. Il faut cependant garder à l'esprit l'étiquetage distant et par conséquent le risque –léger– d'ambiguité sur les étiquettes qui devrait affecter le rappel ainsi que le risque –très fort– d'entités non annotées qui devrait affecter le rappel.

La plupart des ensembles de données de référence en NER présentent un biais de chevauchement entre les termes apparaissant à la fois en entraînement et en test. Ce défaut rend l'interprétation des résultats ambigüe puisque nous ne pouvons pas conclure si le modèle est capable de détecter de nouvelles entités ou si il ne fait que de la mémorisation des entités déjà vues (Taillé *et al.*, 2021).

Afin de limiter ce phénomène, nous proposons de tester nos modèles sur deux ensembles : l'ensemble X composé de phrases nouvelles mais contenant des entités déjà vues en apprentissage (biais de

TABLE 2 – Nombres d'occurences des classes sur les jeux de test X, X_c (hors distribution) ainsi que pour le jeu d'apprentissage

Ensemble	Classe	X	X_c	Jeu d'apprentissage
$\mathcal{Y}_1$	Flower	2988	2800	11242
	Fruit	753	0	3004
	Habit	345	0	1270
	Leaf	618	0	2257
	Part-of	2950	1689	11110
	Stem-root	536	1887	2046
$\mathcal{Y}_2$	Color	1760	5174	6536
	Disposition	929	1210	3537
	Form	2415	745	8630
	Position	1627	0	5932
	Surface-texture	2024	0	7543

chevauchement) et l'ensemble X_c contenant des phrases nouvelles et exclusivement des entitées nommées qui ne figurent pas dans le jeu d'apprentissage. Nous considérons l'ensemble X_c comme un "ensemble de test hors distribution", cet ensemble permettra de mesurer la quantité de nouvelles entités détectées par le modèle. Nous faisons l'hypothèse que cette mesure est très corrélée avec le nombre d'annotations manquantes corrigées par le modèle. Le tableau 2 représente les statistiques des deux jeux de test proposés. Certaines classes ne sont pas représentées dans le jeu de test hors distribution X_c par manque de représentativité dans le jeu de données initial (voir Tableau 1).

4.1 Capacité du modèle à généraliser en fonction du contexte

Dans cette partie les différentes variantes du modèle sont testées sur l'ensemble X qui contient des phrases qui n'ont pas été vues pendant la phase d'apprentissage, mais qui contiennent toujours des entités présentes dans l'ensemble d'apprentissage. Cela nous permet de mesurer la capacité du modèle à s'adapter à de nouveaux contextes et à de nouvelles phrases.

TABLE 3 – Capacité du modèle à détecter et à classifier des entités vues en apprentissage dans un contexte différent (scores en Détection/Classification)

Modèles	Rappel	Précision	Score F1
Baseline	100/92.25	83.87/77.19	91.22/83.07
Baseline w/ lm	100/93.30	84.94/78.99	91.86/84.70
Baseline w/self-train	100/96.34	88.13/84.67	93.69/89.48
Baseline w/ lm_self-train	100/95.30	85.90/81.61	92.41/87.29

Le tableau 3 montre les performances de notre modèle NER avec et sans pré-entraînement du modèle de langue. Nous avons observé une amélioration significative des trois mesures d'évaluation avec le pré-entrainement, y compris une augmentation de 1.63 % du score F1, une augmentation de 1.05 % du rappel et une augmentation de 1.8% du de la précision, en classification. Ces résultats indiquent

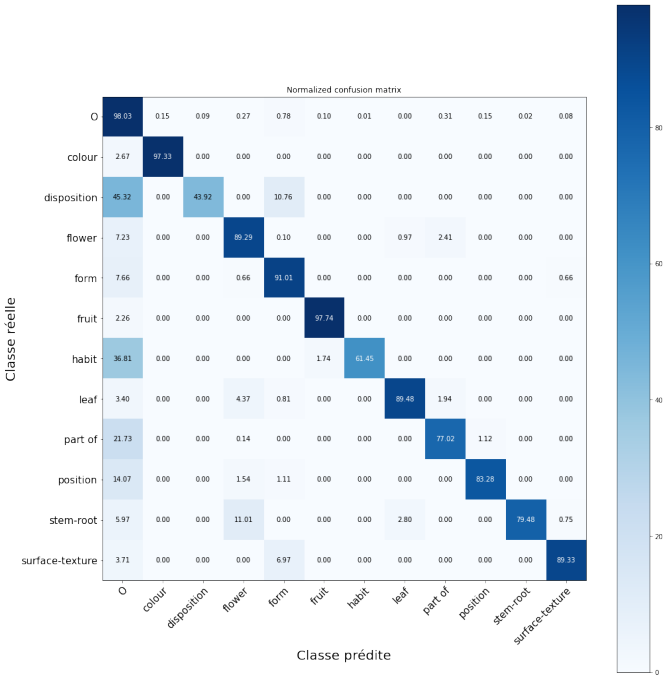


FIGURE 3 – Matrice de confusion du modèle avec auto-apprentissage sur le jeu de données X

que le pré-entraînement du modèle de langue sur un corpus large et diversifié de textes biologiques est une méthode prometteuse pour améliorer les performances du modèle NER sur des données en domaine clos, comme c'est le cas pour les descriptions d'espèces.
Nos expériences sur l'effet de l'auto-apprentissage ont également donné des résultats encourageants. Notre modèle auto-entraîné a atteint un score F1 de 89.48%, surpassant le score F1 du modèle de référence de 4.22%. Ces résultats soulignent l'efficacité de l'auto-apprentissage pour surmonter l'impact négatif de l'annotation distante (bruit et silence sur les étiquettes). Nous pouvons en conclure que l'auto-apprentissage est une technique utile pour améliorer la performance des modèles NER dans les scénarios où les données étiquetées sont rares ou de mauvaise qualité.
Cependant la combinaison d'auto-apprentissage et pré-entrainement du modèle de langue ne semble pas apporter de gain notable sur les performances en détection et en classification, ce qui peut être expliqué par le fait que la combinaison des deux algorithmes conduit à un sur-ajustement aux données.

Sur l'ensemble des expériences, nous notons un écart très significatif entre la détection et la classification. Nous attribuons cet écart au manque de données étiquetées mais nous sommes convaincus qu'il serait possible de le résorber en utilisant par exemple de l'augmentation de données. Il s'agit d'une perspective intéressante pour ce travail.

La matrice de confusion de la figure 3 montre en particulier des erreurs de classification pour la classe **Disposition**, qui a le plus de faux négatifs. Cette classe est souvent confondue avec la classe **O**, la plus représentée du jeu de données d'apprentissage, ainsi qu'avec la classe **Form**, qui est la mieux représentée parmi les classes d'entités. Cependant, il est important de noter que les confusions entre classes de descripteurs et classes d'organes sont très faibles, ce qui indique que le modèle a réussi à assimiler ces notions. Ce dernier point est crucial pour la qualité des triplets extraits par la suite.

FIGURE 4 – Distributions de probabilités du modèle de référence et du modèle avec auto-apprentissage pour la détection d'une entité appartenant à la classe **Part-of**. Jeu de données X_c

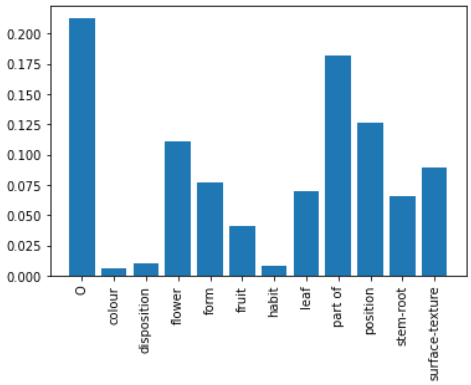


FIGURE 5 – Modèle de référence

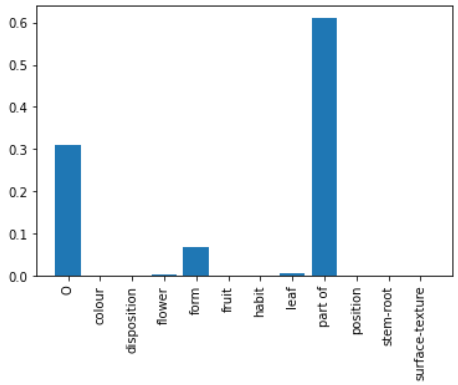


FIGURE 6 – Modèle avec auto-apprentissage

4.2 Capacité des modèles à généraliser sur de nouvelles entités

Le tableau 4 représente les performances des différentes variantes sur l'ensemble X_c qui contient des entités entièrement nouvelles qui n'étaient pas présentes dans l'ensemble d'apprentissage (**hors distribution**). Ce cadre expérimental est très difficile mais il est particulièrement important car il permet d'évaluer la capacité du modèle à détecter et à classifier de nouvelles entités.

TABLE 4 – Capacité des modèles à détecter et à classifier de nouvelles entités, hors de la distribution de l'ensemble d'apprentissage. (Scores en Détection/Classification)

Modèles	Rappel	Précision	Score F1
Baseline	100/71.07	76.79/65.45	86.87/66.62
Baseline w/ lm	100/71.03	83.87/65.16	91.22/66.46
Baseline w/self-train	100/82.11	58.38/47.86	73.75/50.54
Baseline w/ lm_self-train	100/81.54	65.55/55.32	79.19/58.66

Dans le cas où l'ensemble de test ne contient que des entités nommées nouvelles, qui n'ont pas été vues pendant l'apprentissage, le rappel est l'indicateur clé pour mesurer la capacité du modèle à détecter ces nouvelles entités. En effet, le rappel mesure la proportion d'entités nommées prédites par le modèle. Ainsi, la mesure du rappel est plus pertinente que la précision ou le score F1 dans ce contexte particulier.

Dans l'étude mentionnée, le modèle de référence a obtenu un rappel de 71,07% sur l'ensemble de test contenant de nouvelles entités.. Lorsque l'auto-apprentissage a été appliqué au modèle, le rappel a augmenté de manière significative pour atteindre 82.11%. Cependant, le pré-entrainement du modèle de langue n'a pas apporté de gain en rappel. Ce qui démontre l'innefficacité de cette technique pour généraliser sur de nouvelles entités.

Le modèle avec auto-apprentissage est donc celui qui obtient la meilleure précision sur les entités nouvelles.

Qualitativement nous observons sur la figure 4 un exemple d'entité nommée dont la détection a été corrigée par l'auto-apprentissage, l'entité **lobe** appartenant à la classe **Part-of** n'était pas présente dans le jeu d'apprentissage. Nous pouvons clairement observer la baisse d'entropie engendrée par l'auto-apprentissage et son effet bénéfique sur cet exemple.

Il convient de rappeler que tous les modèles ont été entraînés sur un ensemble de données annoté de manière distante, ce qui pourrait entraîner un biais de rétention plus important vers les entités les plus fréquentes. Par conséquent, les améliorations observées dans les modèles avec auto-apprentissage et pré-entraînement du modèle de langue sont encore plus impressionnantes, car elles montrent la capacité de ces techniques à améliorer les performances du modèle sur des entités nouvelles et non vues.

5 Discussion

Dans le contexte des deux ensembles de test, l'un contenant uniquement de nouvelles entités et l'autre contenant de nouvelles phrases avec des entités vues lors de la phase d'apprentissage, il convient de noter que le préentraînement du modèle de langue et l'auto-apprentissage ont tous deux permis d'améliorer les performances du modèle d'extraction d'entités nommées comparé au modèle de référence sur les deux ensembles de données. Cela s'explique par la spécificité du vocabulaire spécialisé et de la forme des phrases. L'auto-apprentissage est plus difficile à régler mais il permet clairement d'augmenter les détections et d'améliorer le rappel au fil des itérations. La question du critère d'arrêt sur cette phase est difficile car la précision n'est pas complètement fiable, un certain nombre de termes pertinents étant probablement étiquetés 0 du fait de l'annotation distante. Il sera nécessaire d'échanger avec les experts du domaine sur la base des prédictions pour trouver le meilleur compromis.

La combinaison du pré-entraînement et de l'auto-entraînement n'a pas permis d'améliorer davantage les performances sur l'un ou l'autre des ensemble de tests et les performances étaient même légèrement inférieures à celles du modèle utilisant uniquement l'auto-apprentissage. En analysant les performances des deux modèles sur de nouvelles entités, les résultats laissent penser qu'une des explication possibles serait que le pré-entrainement du modèle de langue bien qu'apportant une notion de contexte permettant une bonne précision, il ne permet pas généraliser sur des entités nouvelles et combiné à l'auto-apprentissage il risque d'apporter une redondance d'informations et mener à un sur-apprentissage.

Dans l'ensemble, le modèle le plus performant sur les deux ensembles de tests est celui qui a été formé avec l'auto-apprentissage seul. Cela indique que dans le contexte de l'apprentissage supervisé à distance avec des données d'apprentissage limitées, l'auto-apprentissage peut être une technique puissante pour améliorer les performances des modèles de langue dans des contextes de supervision distante et ce même sur des données de domaine clos telles que les descriptions d'espèces.

6 Conclusion

Notre étude visait à améliorer les performances des modèles d'extraction de connaissances pour l'analyse des descriptions d'espèces biologiques. Nous avons proposé un modèle supervisé à distance pour la reconnaissance des entités nommées et décrit un protocole pour la construction de graphes de connaissances à partir de l'étiquetage des entités. Pour évaluer nos modèles avec précision, nous avons proposé un protocole de test consistant en deux ensembles de données, l'un contenant les entités vues pendant l'entraînement et l'autre contenant de nouvelles entités.

Nous avons identifié deux défis scientifiques : la spécificité du vocabulaire et des tournures de phrases, et les annotations manquantes. Pour résoudre le problème du vocabulaire, nous avons proposé une technique de pré-entraînement du modèle de langage qui a amélioré les performances de notre modèle NER le premier ensemble, sans apporter de gain sur le second. Pour le problème des annotations manquantes, nous avons proposé une architecture teacher-student formulée sous forme d'auto-apprentissage, qui a permis d'obtenir le rappel le plus élevé sur les deux ensembles de données.

Pour ce dernier cas de figure, il est d'une part difficile de bien régler ce modèle qui tend à diverger. D'autre part, il est difficile d'interpréter finement les résultats car l'absence d'annotation fiable débouche malheureusement sur différentes interprétations.

En conclusion, notre étude démontre l'apport des modèles de langue récents pour l'analyse de textes complexes et spécifiques. Cette application est vraiment critique pour les chercheurs qui étudient la diversité et l'évolution des espèces, avec des applications potentielles en morphologie comparative et en informatique de la biodiversité. Si l'étude présente ne permet pas de lever tous les verrous scientifiques, elle a contribué largement à inciter les chercheurs du domaine à approfondir ce sujet de recherche : il semble clair aujourd'hui que la construction automatique d'un graphe de connaissances en biologie des espèces est un objectif atteignable à moyen terme.

Au niveau des perspectives, l'auto-apprentissage semble perfectible en introduisant de nouvelles hypothèses et en testant de nouvelles approches. Nous envisageons aussi d'utiliser des modèles de langues plus larges qui semblent encore plus performants en extraction d'entités nommées.

Références

CHO H. & LEE H. (2019). Biomedical named entity recognition using deep neural networks with contextual information. **20**(1), 735. DOI : 10.1186/s12859-019-3321-4.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. Number : arXiv :1810.04805.

JIA C., SHI Y., YANG Q. & ZHANG Y. (2020). Entity enhanced bert pre-training for chinese ner. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6384–6396. DOI : 10.18653/v1/2020.emnlp-main.518.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, p. 260–270.

LEE J., PHAM L. H. & UZUNER O. (2022). Mnlp at fincausal2022 : Nested ner with a generative model. In *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, p. 135–138.

LEITNER E., REHM G. & MORENO-SCHNEIDER J. (2019). Fine-grained named entity recognition in legal documents. In *Semantic Systems. The Power of AI and Knowledge Graphs : 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9–12, 2019, Proceedings*, p. 272–287 : Springer.

LIANG C., YU Y., JIANG H., ER S., WANG R., ZHAO T. & ZHANG C. (2020). BOND : BERT-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 1054–1064 : ACM. DOI : 10.1145/3394486.3403149.

MAGGE A., SCOTCH M. & GONZALEZ-HERNANDEZ G. (2018). Clinical ner and relation extraction using bi-char-lstms and random forest classifiers. In *International workshop on medication and adverse drug event detection*, p. 25–30 : PMLR.

MALOUF R. (2002). Markov models for language-independent named entity recognition. In *COLING-02 : The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

MCCALLUM A. & LI W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, p. 188–191.

MENG Y., ZHANG Y., HUANG J., WANG X., ZHANG Y., JI H. & HAN J. (2021). Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. Number : arXiv :2109.05003.

MIKOLOV T., YIH W.-T. & ZWEIG G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics : Human language technologies*, p. 746–751.

MIWA M. & BANSAL M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1105–1116.

SAUCÈDE T., ELÉAUME M., JOSSART Q., MOREAU C., DOWNEY R., BAX N., SANDS C., MERCADO B., GALLUT C. & VIGNES-LEBBE R. (2021). Taxonomy 2.0 : computer-aided identification tools to assist antarctic biologists in the field and in the laboratory. **33**(1), 39–51. Publisher : Cambridge University Press, DOI : 10.1017/S0954102020000462.

SOUZA F., NOGUEIRA R. & LOTUFO R. (2019). Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv :1909.10649*.

TAI K. S., SOCHER R. & MANNING C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. DOI : 10.48550/arXiv.1503.00075.

TAILLÉ B., GUIGUE V. & GALLINARI P. (2020). Contextualized embeddings in named-entity recognition : An empirical study on generalization. In *Advances in Information Retrieval : 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, p. 383–391 : Springer.

TAILLÉ B., GUIGUE V., SCOUTHEETEN G. & GALLINARI P. (2021). Separating retention from extraction in the evaluation of end-to-end relation extraction. Number : arXiv :2109.12008.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

VIGNES-LEBBE R., BOUQUIN S., KERNER A. & BOURDON E. (2017). Desktop or remote knowledge base management systems for taxonomic data and identification keys : Xper2 and xper3. **1**, e19911. Publisher : Sofia : Pensoft Publishers, 2017-, DOI : 10.3897/tdwgproceedings.1.19911.

WANG X., HU V., SONG X., GARG S., XIAO J. & HAN J. (2021). Chemner : fine-grained chemistry named entity recognition with ontology-guided distant supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

WANG X., SONG X., LI B., ZHOU K., LI Q. & HAN J. (2020). Fine-grained named entity recognition with distant supervision in covid-19 literature. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 491–494 : IEEE.

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

Le théâtre français du XVIIe siècle : une expérience en catégorisation de textes

Jacques Savoy
Institut d'informatique, Université de Neuchâtel, Suisse
Jacques.Savoy@unine.ch

RÉSUMÉ

La catégorisation de documents (attribution d'un texte à une ou plusieurs catégories prédéfinies) possède de multiples applications. Cette communication se focalise sur l'attribution d'auteur en analysant le style de vingt pièces de théâtre du XVIIe siècle. L'hypothèse que nous souhaitons vérifier admet que le véritable auteur est le nom apparaissant sur la couverture. Afin de vérifier la qualité de deux méthodes d'attribution, nous avons repris deux corpus additionnels basés sur des romans écrits en français et italien. Nous proposons une amélioration de la méthode Delta ainsi qu'une nouvelle grille d'analyse pour cette approche. Ensuite, nous avons appliqué ces approches sur notre collection de comédies. Les résultats démontrent que l'hypothèse de base doit être écartée. De plus, ces œuvres présentent des styles proches rendant toute attribution difficile.).

ABSTRACT

The French Theater of the 17th Century : An Experiment in Text Categorization

The automatic assignment of a text to one or more predefined categories presents multiple applications. In this context, the current study focuses on author attribution in which the true author of a doubtful text must be identified. This analysis emphasis on the style of 20 French comedies written in verse by 9 authors during the 17th century. The hypothesis we want to verify assumes that the real author is the name appearing on the cover (called the signature hypothesis). In order to validate the reliability of two attribution procedures, we used two additional corpora based on 200 extracts of novels written in French with 30 authors and 140 Italian novels authored by 40 persons. After this verification, we propose an improvement of the Delta method as well as a new analysis grid for this model. Finally, we applied these approaches to our French comedy corpus. The results demonstrate that the signature hypothesis must be discarded. Moreover, these works present similar styles making any attribution difficult to support with a high degree of certainty.

MOTS-CLÉS : Classification automatique, humanités numériques, apprentissage automatique, attribution d'auteur. .

KEYWORDS: Text Categorization, digital humanities, machine learning, authorship attribution.

1 Introduction

Depuis plus de 150 ans, l'identité du véritable auteur d'Hamlet (écrit en 1601) ou de Roméo et Juliette (1594) a fait l'objet de quelques 4 000 livres et articles (Michell, 1999). Plusieurs noms ont été proposés comme F. Bacon, C. Marlowe, E. de Vere et pour les plus récents, J. Florio (Tassinari, 2009). Face à ces nombreuses propositions, les Stratfordiens récusent toute autre attribution que celle

de Shakespeare. Ce débat n'est pas clos et toute apparition d'une œuvre pouvant être attribuée à Shakespeare fait renaître la discussion (Kreuz, 2023).

Pour le XVIe ou XVIIe siècle, toute attribution s'avère complexe en raison de l'absence d'un véritable droit d'auteur et par la possibilité d'avoir un (ou deux) co-auteur(s) pour une pièce (Craig & Kinney, 2009). Dans le monde francophone, le débat s'est focalisé sur le tandem Molière – P. Corneille. L'objectif de cette communication n'est pas de trancher cette question mais d'appliquer les méthodes d'attribution d'auteur les plus fiables sur les comédies du XVIIe siècle (en excluant celles attribuées à Molière ou P. Corneille).

En présence d'une pièce de théâtre, on peut supposer que son véritable auteur correspond au nom imprimé sur la couverture (affirmation que nous nommerons hypothèse de la signature). Pour vérifier cette thèse, nous avons repris vingt comédies en alexandrins écrites par neuf auteurs (J. G. deCampistron (1656—1723), Champmeslé (1642—1701), Chevalier (16..—1673), Hauteroche (1617—1707), Montfleury (1608—1667), P. Quinault (1635—1688), J. Racine (1639-–1699), Tristan (1601—1655), et T. Corneille (1625—1709)). Si cette hypothèse se vérifie, nous devrions distinguer neuf écrivains présentant des styles dissemblables. Dans le cas contraire, l'hypothèse de la signature doit être abandonnée et nous devrions identifier le véritable auteur pour chaque œuvre.

Dans cette étude, nous supposons que les comédies retenues ont été écrites par un seul auteur, ce qui est généralement le cas. On peut rencontrer de temps à autre des exceptions, comme Psyché (1671) rédigé par P. Corneille, Molière et P. Quinault (selon l'hypothèse de la signature). De plus, l'auteur est la personne qui a rédigé le texte et non celle qui a fourni l'intrigue, des dialogues, des scènes comiques ou financé la rédaction de l'œuvre.

Dans la suite de cet article, nous présenterons un survol des connaissances en attribution d'auteur (section 2). La troisième section décrit nos trois corpus. La quatrième explique l'attribution fondée sur une distance intertextuelle et la cinquième expose la méthode Delta et nos améliorations. Enfin, une dernière section applique ces deux approches sur les vingt pièces de théâtre français. Une conclusion dresse les principaux résultats de cette étude.

2 État des connaissances

En catégorisation de textes (Sebastiani, 2002), on distingue entre les modèles basés sur la sémantique (e.g., indexation automatique, filtrage, etc.) ou le style (Savoy, 2020), (Karsdorp *et al.*, 2021). Dans ce dernier cas, les applications en stylométrie couvrent un champ assez large, allant de l'attribution et profilage d'auteur, à la détection de faux ou de plagiat, support en criminologie (Olsson, 2018), voire la datation d'un document (Kreuz, 2023).

Les premières approches datant du XIXe siècle (Mendenhall, 1887) ont proposé des mesures simples comme la longueur moyenne des mots, ou le pourcentage de termes apparaissant une seule fois (*hapax*) afin d'identifier différents styles. Toutefois, l'instabilité de ces mesures face à des textes de longueur variable rend ces solutions inopérantes (Baayen, 2008).

En se limitant à la question de l'identification de l'auteur, trois contextes sont possibles. Premièrement, dans un environnement fermé, le véritable auteur est l'un des écrivains proposés et, pour chaque candidat un ensemble de textes est fourni. Deuxièmement, l'auteur peut être un des noms mentionnés ou un autre, encore inconnu (contexte ouvert). Troisièmement, le système doit répondre si deux

textes ont été rédigés par la même plume ou non. Dans ces trois situations, une réponse ne saurait se limiter à un simple nom, et une justification plus complète devrait être fournie. Au cours de ces trois dernières décennies, des approches plus performantes ont été proposées que l'on peut classifier selon les attributs stylistiques retenus d'une part et, d'autre part, les mesures de similarité (ou de distance) appliquées.

Dans une première grande famille, on peut regrouper les modèles s'appuyant sur un ensemble de vocables sélectionnés. Ces derniers peuvent être fréquemment employés de manière inconsciente et correspondent à des mots-outils comme les articles (le, des), pronoms (tu, nous), prépositions (sur, vers), conjonctions (et, mais) et des verbes auxiliaires et modaux (est, avait, fait) (Hughes *et al.*, 2012). Cette énumération correspond à un anti-dictionnaire (*stoplist*) usité habituellement par les moteurs de recherche. D'autres modèles suggèrent de tenir compte des termes relativement fréquents chez un auteur et peu ou pas employés par les autres (Burrows, 2007), (Craig & Kinney, 2009). Finalement, certains modèles se fondent sur l'ensemble du vocabulaire, parfois en éliminant certains vocables peu fréquents (Labbé, 2007) ou en laissant le modèle sélectionner de manière automatique les termes les plus pertinents.

Avec l'accroissement des capacités des ordinateurs et les campagnes d'évaluation CLEF-PAN sur ce thème (Rosso *et al.*, 2019), la distinction entre différents styles peut s'appuyer sur de brèves séquences de lettres (n-grammes, avec $n = 2$ à 6) (Kjell, 1994). Évidemment, la combinaison des mots et des chaines de lettres permet de fournir un grand nombre d'attributs stylistiques (Savoy, 2020).

La détermination du style propre d'un auteur peut également s'établir en considérant la syntaxe comme, par exemple, en se fondant sur de courtes séquences de mots, soit les plus fréquentes, soit celles respectant des patrons prédéfinis (e.g., adjectif-nom-nom) (Kocher & Savoy, 2019). Comme autre source d'information, la longueur moyenne des phrases ou leur distribution peuvent fournir des attributs complémentaires et discriminants.

Dès que chaque texte est représenté par une liste d'attributs, les modèles d'attribution peuvent se baser sur une fonction de distance (ou similarité) entre textes ou recourir à un modèle d'apprentissage automatique (e.g., SVM, les k plus proches voisins) voire par un apprentissage profond. Dans tous les cas, l'attribution s'établit selon la règle du voisin le plus similaire (ou de la distance la plus faible). Signalons toutefois que toute stratégie basée sur l'apprentissage par machine requiert un jeu d'entrainement, données qui ne sont pas toujours disponibles. De plus, une forte corrélation doit exister entre le jeu d'apprentissage et celui du test. Ces contraintes impliquent que le contexte d'attribution soit fermé (le véritable auteur doit être présent dans le jeu d'apprentissage). Dans le cas contraire (contexte ouvert), la réponse proposée risque d'être erronée.

3 Les trois corpus étudiés

Afin d'analyser quelques comédies du XVIIe siècle, nous avons recouru à trois corpus différents. Nos deux premiers corpus serviront à vérifier la qualité des deux méthodes retenues pour l'attribution d'auteur. La première collection se compose de 150 romans contemporains écrits en italien par 40 auteurs distincts. Ce corpus nommé PIC (*Padova Italian Corpus*) a été construit par un groupe de chercheurs à l'Université de Padoue sous la supervision du prof. Cortelazzo et du prof. Tuzzi (Tuzzi & Cortelazzo, 2018).

La table 3 (dans les annexes) indique le nom des auteurs, le sexe et le nombre de romans inclus

dans le corpus. On y retrouve 27 hommes et 12 femmes de même que le nom E. Ferrante dont on souhaite découvrir la véritable identité. Tous les romans correspondent au même genre, soit des textes pour adultes. Tous les éléments n'appartenant pas au récit ont été soigneusement éliminés (e.g., numérotation des pages, titre courant, etc.). Le roman le plus long comprend 196 914 formes (Faletti, *Io uccito*, 2002) et le plus bref seulement 7 694 formes (Parrella, *Behave*, 2011, le seul document ayant moins de 10 000 formes).

Dans l'évaluation d'une procédure d'attribution d'auteur, il est important de souligner trois contraintes. D'abord, chaque document doit être assez long. Dans ce corpus italien, chaque test contient 10 000 formes ou plus (à une exception près). Ensuite, l'orthographe a été vérifiée. Enfin, d'autres facteurs pouvant influencer le style doivent être réduits au strict minimum. Ainsi cette collection renferme des textes écrits dans la même langue, et rédigés durant la même période (de 1987 à 2016).

Le deuxième corpus nommé St-Jean comprend 200 extraits de 67 romans écrits en français par 30 auteurs différents (Labbé, 2017). La distribution entre les divers auteurs est indiquée dans la table 4 (voir annexes). Chaque document comprend approximativement 10 000 formes. La plage temporelle couverte par ce corpus s'étend sur tout le XIXe siècle de Chateaubriand (*Atala*, 1801) à Proust (*Les Plaisirs et les jours*, 1896). Ce corpus respecte également les contraintes citées ci-dessus.

Le troisième corpus comprend une sélection de vingt pièces de théâtre dont la table 5 décrit les principales caractéristiques (voir annexes). Ces œuvres sont toutes rédigées en français et correspondent à des comédies en vers. Elles contiennent en général plus de 10 000 formes et couvrent la période de 1651 à 1709.

4 La distance intertextuelle

Afin de déterminer si deux documents sont rédigés par le même auteur, nous comparons l'ensemble du vocabulaire (Labbé, 2007). Si les termes employés et leur fréquence s'avèrent proches, la distance intertextuelle sera faible. Dans le cas contraire, elle s'élèvera jusqu'à un maximum de 1,0 lorsque deux textes ne possèdent rien en commun comme un roman écrit en français et un autre en finnois. A l'inverse, si les deux documents sont identiques, la distance sera nulle.

Plus précisément, la distance intertextuelle entre le texte A et B (notée $D(A,B)$) est indiquée dans l'équation 1 dans laquelle n_A signale le nombre de mots (*tokens*) du texte A et tf_{iA} la fréquence absolue du terme i (pour $i = 1, 2, \ldots, m$) dans le texte A. La taille du vocabulaire est indiquée par m. Si l'on admet que le texte B est plus long que le texte A, nous devons réduire les fréquences des termes appartenant à B. Ces dernières (notées tf_{iB}) sont multipliées par le rapport des tailles comme présenté à droite la formule 1.

$$D(A,B) = \frac{\sum_{i=1}^{m} |tf_{iA} - \hat{tf}_{iB}|}{2 \cdot n_A} \qquad avec\ \hat{tf}_{iB} = tf_{iB} \cdot \frac{n_A}{n_B} \tag{1}$$

La machine calcule la distance entre toutes les paires de romans présents dans un corpus (soit 200 x199 / 2 = 19 900 pour la collection St-Jean). Pour chaque œuvre, on peut alors déterminer les k extraits les plus proches.

Avec le corpus St-Jean, le plus proche voisin de chaque extrait correspondait toujours à un passage d'un roman rédigé par le même auteur. Une distance intertextuelle inférieure à 0,2 constitue une très

forte évidence que les deux textes sont du même auteur (Labbé, 2007), (Savoy, 2018). Parmi les facteurs pouvant favoriser une faible distance on peut ajouter la présence de texte étant du même genre et écrit dans la même décennie. La précision est donc de 100 % avec la règle empirique de 0,2 ; en d'autres termes, si la distance est inférieure à 0,2, les deux documents sont de la même plume. Cependant, deux textes du même auteur peuvent présenter une distance supérieure à cette limite. Le taux de rappel n'est donc pas parfait.

Avec la collection PIC dans laquelle on compare tout un roman avec un autre, le résultat s'avère similaire sauf pour certains romans d'Elena Ferrante. Pour être précis, seulement les trois derniers romans d'Elena Ferrante présentent un appariement inférieur à 0,2 avec trois romans de D. Starnone (soit *Autobiografia, Lacci* ou *Schezetto*).

En analysant les autres écrivains italiens, on constate généralement que les romans d'un même auteur présentent des distances supérieures à 0,2 (Savoy, 2018). On peut avancer l'hypothèse qu'un écrivain cherche souvent d'autres perspectives ou souhaite aborder d'autres thèmes avec un ton quelque peu différent. Toutefois, notre corpus italien indique que pour trois auteurs (Carofiglio, Faletti ou Veronesi), les distances entre romans demeurent souvent inférieures à 0,2.

5 Delta

Afin d'identifier le style d'un texte, le modèle Delta tient compte des m vocables (ou lemmes) les plus fréquents employés par les auteurs du corpus étudié (Burrows, 2002). L'opérateur est libre de fixer la valeur de m, mais les plus courantes varient entre 50 et 500. L'idée sous-jacente consiste à tenir compte des mots fréquents étant peu ou pas porteurs de sens et utilisés de manière inconsciente par l'auteur. Avec une valuer de m inférieure à 300 ou 400, la large majorité d'entre eux sont des mots-outils qui s'avèrent indépendant des thèmes du texte et donc plus associés au style de l'auteur.

Chaque terme t_i sélectionné possèdera un poids dénoté $Zscore(t_{ij})$ correspondant à la différence entre sa fréquence relative dans le texte T_j (notée rtf_{ij}) et la moyenne ($\overline{rtf}_i$) pour ce terme t_i sur l'ensemble des textes du corpus. Afin de tenir compte de la variabilité sous-jacente, chaque différence est divisée par l'écart-type (s_i).

$$Z\ score(t_{ij}) = \frac{rtf_{ij} - \overline{rtf}_{ij}}{s_i} \tag{2}$$

Étant donné un texte Q dont l'attribution est incertaine et un texte A_j (écrit par l'auteur A_j), nous pouvons calculer la différence en valeur absolue entre les scores Z du texte Q et A_j et d'en calculer la moyenne (voir équation 3).

$$\Delta(Q, A_j) = \frac{1}{m} \cdot \sum_{i=1}^{m} |Z\ score(t_{iQ}) - Z\ score(t_{iA_j})| \tag{3}$$

dans laquelle m indique le nombre de termes sélectionnés et t_{iA_j} le ie terme dans le texte A_j (de même pour t_{iQ} et le texte Q). Dans nos expériences, la valeur de m a été fixée à 200. Ce choix s'explique par notre souci de baser notre similarité stylistique sur les mots-outils et non des termes

TABLE 1 – Distribution des probabilités dans un contexte ouvert (“O”) ou fermé (“F”)

	1	2	3	4	5	6	7	8	9	10
O	30,7%	16,0%	10,6%	8,0%	6,0%	4,8%	3,9%	3,3%	2,7%	2,3%
F	94,4%	1,8%	1,0%	0,6%	0,4%	0,3%	0,2%	0,2%	0,2%	0,1%

liés aux thèmes des pièces de théâtre. Comme m varie d'une application à l'autre, on ne peut pas calibrer les distances retournées comme la distance intertextuelle le permettait.

Souvent appliquée en stylométrie (Karsdorp *et al.*, 2021), cette approche permet d'estimer une distance entre des textes, avec la plus faible valeur indiquant l'auteur probable du document Q. Afin d'estimer une probabilité que la distance obtenue indique le véritable auteur, nous proposons d'appliquer la fonction $softmin()$ sur l'ensemble des distances calculées comme exprimée dans l'équation 4.

$$Prob(A_j) = \frac{e^{-c \cdot \Delta(Q, A_j)}}{\sum_{i=1}^{k} e^{-c \cdot \Delta(Q, A_i)}} \tag{4}$$

dans laquelle $Prob(A_j)$ signale la probabilité que le texte Q ait été rédigé par A_j. La constante c, fixée à 20 dans nos expériences, permet de mieux disperser les valeurs retournées par la fonction Delta.

Ce modèle admet implicitement que le véritable auteur se trouve parmi les écrivains proposés (contexte fermé). Si ce n'est pas le cas, Delta classera tout de même tous les auteurs présents selon leur similarité avec le document cible. Afin de vérifier les divergences entre les valeurs retournées dans ces deux contextes, nous avons repris nos extraits de romans écrits en français. Par itération, on a recherché l'auteur de chaque extrait. Dans une première expérience, le véritable auteur a toujours été éliminé (contexte ouvert). En observant les valeurs Delta retournées, on constate que les intervalles de distance entre les cinq premiers rangs demeurent relativement faibles.

En transformant ces distances en probabilités, nous pouvons extraire une distribution des probabilités pour le contexte ouvert (“O”) ou fermé (“F”) (voir la table 1). Dans le premier cas (étiquette “O”), on constate que l'attribution au plus proche voisin signale une probabilité moyenne de 30,7 % pour le premier rang, 16 % pour le deuxième et 10,6 % pour le troisième. En sommant ces trois valeurs, on obtient un total de 57,3 %. En considérant les rangs suivants, les probabilités décroissent lentement. Comme le véritable auteur est absent, on comprend que cette distribution de probabilités se disperse sur les dix ou quinze premiers.

Par contre, si le véritable auteur est présent (contexte fermé, ligne “F”), la probabilité estimée pour le premier rang s'élève à 94,4 %. La différence entre le premier et le deuxième rang s'avère très nette. Des histogrammes similaires peuvent être générés depuis le corpus italien.

Bien que l'application directe de la méthode Delta ne puisse apporter une attribution claire, le recours à une estimation des probabilités permet de résoudre ce problème.

TABLE 2 – Distance inférieure 0,2 entre des pièces d'auteurs différents

AuteurTitre	AuteurTitre	AuteurTitre	AuteurTitre	AuteurTitre
CamJal	MonFem			
ChaPar				
ChaRag				
CheCar				
ChePéd	HauMus			
HauAma	MonFem	MonCom	QuiMèr	TCoDom/Alb/Fes/Com
HauMus	ChePéd	MonCom	QuiMèr	TCoAlb/Fes
MonFem	CamJal	HauAma	QuiMèr	TCoDom/Amo/Gal/Alb/Fes
MonCom	HauAma	HauMus		TCoCom/Alb/Fes
QuiRiv				
QuiMèr	HauAma	HauMus	MonFem	TCoAmo/Alb/Com/Fes
RacPla				
TriAma				
TriPar				
TCoDom	HauAma	MonFem		
TCoAmo	MonFem	QuiMèr		
TCoGal	MonFem			
TCoAlb	HauAma	HauMus	MonFem	MonCom QuiMèr
TCoCom	HauAma	MonCom	QuiMèr	
TCoFes	HauAma	HauMus	MonFem	MonCom QuiMèr

6 Evaluation

Reprenons notre corpus de vingt comédies du théâtre français. Notre analyse stylométique s'effectuera pièce par pièce sans supposer que toutes les œuvres parues sous le même auteur sont effectivement écrites par la même plume.

Afin de pouvoir présenter nos résultats, nous désignons chaque pièce par les trois premières lettres de son auteur (e.g., "Hau" pour Hauteroche) suivies des trois premières lettres du mot principal dans le titre (e.g., "Ama" pour *L'amant qui ne flatte point*). Comme notre corpus contient six œuvres de T. Corneille et que ces dernières sont souvent proches des autres pièces, nous ne répèterons pas le nom de T.Corneille (ou "TCo") pour chaque comédie. Ainsi l'acronyme "TCoAlb/Fes" indique les deux pièces de T. Corneille *Le baron d'Albikrac* et *Le festin de Pierre*.

La table 2 indique les rapprochements stylistiques déduits avec la méthode de la distance intertextuelle. Dans cette table, les similarités provenant de comédies du même auteur sont ignorées. Si l'hypothèse de la signature se confirme, la table 2 doit être vide. Pour chaque texte, aucune forte similarité stylistique ne devrait exister avec une comédie écrite par un autre auteur. Dans la table 2, certaines comédies ne se rapprochent d'aucune autre comme les deux pièces attribuées à Champmeslé, une à Chevalier (*L'intrigue des carrosses*), une de P. Quinault (*Les rivales*), deux à Tristan et *Les plaideurs* de J. Racine. Selon la distance intertextuelle, ces pièces respectent l'hypothèse de la signature.

A l'inverse, les deux pièces de Hauteroche ou de Montfleury s'apparentent fortement à des comédies d'Hauteroche, Montfleury, P. Quinault et T. Corneille. Avec J. G. de Campistron, l'unique comédie retenue (*Le jaloux désabusé*) se rapproche d'une pièce de Montfleury (*La femme juge et partie*). Pour

la comédie *Le pédagogue amoureux* de Chevalier, on observe également une similitude stylistique avec une comédie signée Hauteroche (*Crispin musicien*).

Dans les grandes lignes, la table 2 signale une forte similarité entre les comédies de Hauteroche, Montfleury, T. Corneille et une pièce de Quinault. Est-ce que ce rapprochement provient des thèmes et partiellement du style ?

Afin d'exclure une forte influence des thèmes, nous avons appliqué l'approche Delta sur l'ensemble des 13 comédies présentant un rapprochement stylistique. Les résultats obtenus confirment une similarité stylistique entre ces pièces. Par exemple, la moyenne des probabilités associées au premier rang pour ces 13 comédies s'élève à 38,7% et à 21% pour la seconde position. La méthode Delta n'arrive pas à identifier une plume unique pour chacune de ces œuvres.

En considérant P. Quinault et Chevalier, on remarque que pour ces deux auteurs une comédie présente de forte similarité stylistique avec d'autres œuvres tandis que la seconde propose un style original. Cette constatation justifie une analyse texte par texte au lieu de travailler avec un profil d'auteur généré depuis l'ensemble de ses œuvres. Toutefois cette absence de similitude entre pièces peur aussi s'expliquer par le choix restreint de cette étude. Seulement vingt comédies ont été analysées. De plus, d'autres auteurs n'ont pas été repris.

7 Conclusion

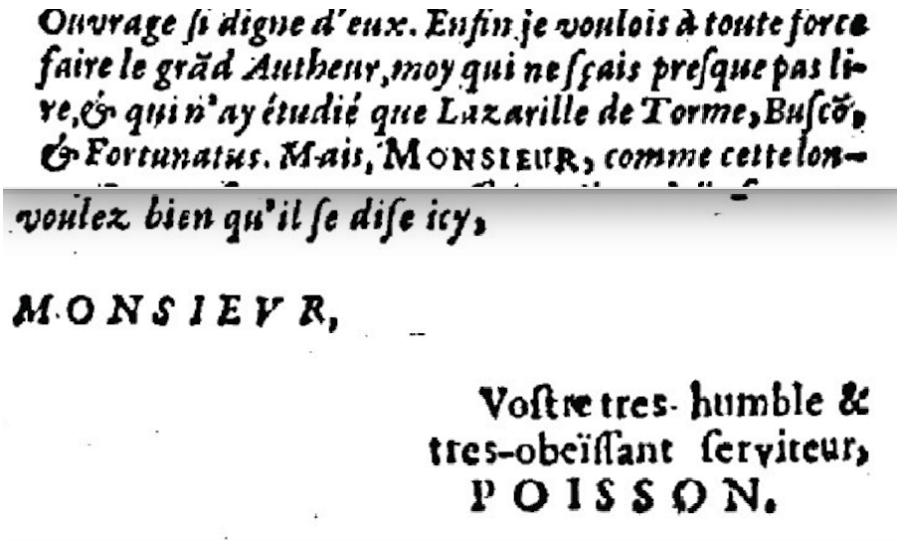
Cette étude explore les similarités stylistiques entre vingt comédies françaises écrites au XVIIe siècle par neuf auteurs distincts. Dans un premier temps, nous avons admis l'hypothèse de la signature ; le véritable auteur s'avère celui dont le nom apparait sur la couverture. Afin de vérifier cette affirmation, nous avons appliqué deux méthodes d'attribution d'auteur soit la distance intertextuelle (méthode non-supervisée) (Labbé, 2007) et l'approche Delta(Burrows, 2002).

Afin d'adapter cette dernière à un contexte ouvert (le véritable auteur demeure inconnu), nous proposons d'estimer une probabilité d'attribution sur la base des distances retournées par le modèle Delta. Avec cette estimation, nous observons clairement deux distributions distinctes de probabilités, à savoir lorsque le véritable auteur est présent ou absent du jeu d'entrainement. Avec cet ajustement, la méthode Delta s'applique à un contexte ouvert (le véritable auteur demeurant inconnu) et l'interprétation des distances s'en trouve simplifié.

Deuxièmement, nous avons testé l'hypothèse de la signature. Comme l'indique la table 2, cette dernière doit être écartée. Certes, pour certains écrivains (e.g., J. Racine, Tristan, Champmeslé), aucune similarité importante avec d'autres textes n'a été détectée. Pour d'autres, le style de certaines comédies s'apparente fortement au style d'œuvres parues sous le nom d'autres auteurs. En particulier, ce rapprochement touche des textes attribués à Hauteroche, Montfleury, P. Quinault et T. Corneille. Si nous n'avons pas quatre auteurs, combien en avons-nous réellement ? Doit-on pencher vers un auteur unique qui pourrait être T. Corneille ?

Avec les approches proposées, cette étude ne permet pas d'identifier sans l'ombre d'un doute le véritable auteur ou d'indiquer si certaines de ces comédies ont été écrites par deux (ou plusieurs) écrivains. Enfin, notre démarche s'appuie uniquement sur les textes et ignore les évidences externes (comme, par exemple, une étude de biographie comparée des auteurs, une concordance des dates, un livre de compte (Young & Young, 1977), ...) pouvant confirmer ou infirmer une possible attribution.

Un exemple d'évidence externe explicite est repris dans la figure 1. Dans cette préface, l'auteur (R. Poisson) indique qu'il sait "presque pas lire", un exemple complémentaire qui signale que l'hypothèse de la signature est infirmée.


Ouvrage ſi digne d'eux. Enfin je voulois à toute force
faire le grãd Autheur, moy qui ne ſçais preſque pas li-
re, & qui n'ay étudié que Lazarille de Torme, Buſcõ,
& Fortunatus. Mais, MONSIEUR, comme cette lon-
voulez bien qu'il ſe diſe icy,

MONSIEVR,

Voſtre tres-humble &
tres-obeïſſant ſerviteur,
POISSON.

FIGURE 1 – Extrait de la préface du *Le Poète basque* (1668) de R. Poisson

Remerciements

Le corpus Ferrante a été créé par les professeurs A. Tuzzi et M. Cortelazzo (Tuzzi & Cortelazzo, 2018) de l'Université de Padoue. Le corpus St-Jean a été construit par D. Labbé (Labbé, 2017).

Références

BAAYEN H. (2008). *Analysis Linguistic Data : A Practical Introduction to Statistics Using R*. Cambridge University Press, Cambridge.

BURROWS J. (2002). Delta : A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, **17**(3), 267–287.

BURROWS J. (2007). All the way through : Testing for authorship in different frequency strata. *Literary and Linguistic Computing*, **22**(1), 27–47.

CRAIG H. & KINNEY A. (2009). *Shakespeare, Computers, and the Mystery of Authorship*. Cambridge University Press, Cambridge.

HUGHES J., FOTI N., KRAKAUER D. & ROCKMORE D. (2012). Quantitative patterns of stylistic influence in the evolution of literature. *Proceedings of the National Academy of Science (PNAS)*, **109**(20), 7682–7686.

KARSDORP F., KESTEMONT M. & RIDDELL A. (2021). *Humanities Data Analysis. Case Studies with Python*. Princeton University Press, Princeton.

KJELL B. (1994). Authorship determination using letter pair frequency features with neural network classifier. *Literary and Linguistics Computing*, **9**(2), 119–124.

KOCHER M. & SAVOY J. (2019). Evaluation of text representation schemes and distance measures for authorship linking. *Digital Scholarship in the Humanities*, **34**(1), 189–207.

KREUZ R. (2023). *Linguistics Fingerprints. How Language Creates and Reveals Identity*. Prometheus Books, Guilford.

LABBÉ D. (2007). Experiments on authorship attribution by intertextual distance in English. *Journal of Quantitative Linguistics*, **14**(1), 33–80.

LABBÉ D. (2017). *Une expérience d'attribution d'auteur. Le corpus St-Jean*. Rapport interne, Université de Grenoble.

MENDENHALL T. (1887). The characteristic curves of composition. *Science*, **214**, 237–249.

MICHELL J. (1999). *Who Wrote Shakespeare*. Thames and Hudson, London.

OLSSON J. (2018). *More Wordcrime. Solving Crime Through Forensic Linguistics*. Bloomsbury, London.

ROSSO P., POTTHAST M., STEIN B., STAMATATOS E., RANGEL F. & DAELEMANS W. (2019). *Evolution of the PAN Lab on Digital Text Forensics*, In *Information Retrieval Evaluation in a Changing World*, p. 461–486. Springer, Cham.

SAVOY J. (2018). Is Starnone really the author behind Ferrante ? *Digital Scholarship in the Humanities*, **33**(4), 902–918.

SAVOY J. (2020). *Machine Learning Methods for Stylometry : Authorship Attribution and Author Profiling*. Springer, Cham.

SEBASTIANI F. (2002). Machine learning in automated text categorization. *ACM Computing Survey*, **14**(1), 1–27.

TASSINARI L. (2009). *John Florio, The Man who was Shakespeare*. Giano Books, New York.

TUZZI A. & CORTELAZZO M. (2018). What is Elena Ferrante ? A comparative analysis of a secretive bestselling Italian writer. *Digital Scholarship in the Humanities*, **33**(3), 685–702.

YOUNG B. & YOUNG G. (1977). *Le Registre de La Grange 1659-1685*. Slatkine, Genève.

Annexe

TABLE 3 – Nom des auteurs, sexe (H/F), et nombre de romans

Nom	Sexe	Nombre	Nom	Sexe	Nombre
Affinati	H	2	Montesano	H	2
Ammaniti	H	4	Morazzoni	F	2
Bajani	H	3	Murgia	F	5
Balzano	H	2	Nesi	H	3
Baricco	H	4	Nori	H	3
Benni	H	3	Parrella	F	2
Brizzi	H	3	Piccolo	H	7
Carofiglio	H	9	Pincio	H	3
Covacich	H	2	Prisco	H	2
De Luca	H	4	Raimo	H	2
De Silva	H	5	Ramondino	F	2
Faletti	H	5	Rea	H	3
Ferrante	?	7	Scarpa	H	4
Fois	H	3	Sereni	F	6
Giordano	H	3	Starnone	H	10
Lagioia	H	3	Tamaro	F	5
Maraini	F	5	Valerio	F	3
Mazzantini	F	4	Vasta	H	2
Mazzucco	F	5	Veronesi	H	4
Milone	F	2	Vinci	F	2

TABLE 4 – Nom des auteurs, nombre d'extraits et nombre de romans

Nom	Extrait	Roman	Nom	Extrait	Roman
Balzac	13	6	Maupassant	10	5
Barbey	8	2	Musset	3	1
Bourget	6	1	Nerval	5	2
Chateaubriand	3	2	Proust	3	1
Daudet	6	1	Régnier	8	2
Dumas	10	3	Sainte-Beuve	6	1
Erckmann	4	1	Sand	10	3
Flaubert	11	6	Staël	6	1
France	8	2	Stendhal	6	2
Fromentin	5	1	Sue	10	1
Gautier	5	3	Vallès	4	1
Goncourt	5	2	Verne	4	2
Huysmans	4	2	Victor	8	2
Lamartine	7	2	Vigny	5	2
Loti	7	2	Zola	10	5

TABLE 5 – Nom des auteurs, titre des comédies, année de parution et longueur

Auteur	Titre	Année	Formes
Campistron	*Le jaloux désabusé*	1709	14 383
Champmeslé	*Le Parisien*	1684	18 091
Champmeslé	*Ragotin*	1684	15 596
Chevalier	*L'intrigue des carrosses*	1662	10 154
Chevalier	*Le pédagogue amoureux*	1665	16 977
Hauteroche	*L'amant qui ne flatte point*	1668	20 908
Hauteroche	*Crispin musicien*	1671	22 350
Montfleury	*La femme juge et partie*	1669	17 464
Montfleury	*Le comédien poète*	1673	21 771
Quinault	*Les rivales*	1653	18 680
Quinault	*La mère coquette*	1665	19 452
Racine	*Les plaideurs*	1668	10 063
Tristan	*Amaryllis*	1652	16 124
Tristan	*Le parasite*	1654	18 701
T. Corneille	*Dom Bertran de Cigarral*	1651	20 911
T. Corneille	*L'amour à la mode*	1651	20 819
T. Corneille	*Le galant doublé*	1659	21 152
T. Corneille	*Le baron d'Albikrac*	1667	20 558
T. Corneille	*La comtesse d'Orgueil*	1670	21 124
T. Corneille	*Le festin de Pierre*	1677	20 068

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

Enrichissement des modèles de langue pré-entraînés par la distillation mutuelle des connaissances

Raphaël Sourty[♠◇] Jose G. Moreno[♠] François-Paul Sevant[◇] Lynda Tamine[♠]
[♠]Université Paul Sabatier, IRIT, UMR 5505 CNRS, Toulouse, France
[◇]Renault, Boulogne-Billancourt, France

RÉSUMÉ

Les bases de connaissances sont des ressources essentielles dans un large éventail d'applications à forte intensité de connaissances. Cependant, leur incomplétude limite intrinsèquement leur utilisation et souligne l'importance de les compléter. À cette fin, la littérature a récemment adopté un point de vue de monde ouvert en associant la capacité des bases de connaissances à représenter des connaissances factuelles aux capacités des modèles de langage pré-entraînés (PLM) à capturer des connaissances linguistiques de haut niveau et contextuelles à partir de corpus de textes. Dans ce travail, nous proposons un cadre de distillation pour la complétion des bases de connaissances où les PLMs exploitent les étiquettes souples sous la forme de prédictions d'entités et de relations fournies par un modèle de plongements de bases de connaissances, tout en conservant leur pouvoir de prédiction d'entités sur de grandes collections des textes. Pour mieux s'adapter à la tâche de complétion des connaissances, nous étendons la modélisation traditionnelle du langage masqué des PLM à la prédiction d'entités et d'entités liées dans le contexte. Des expériences utilisant les tâches à forte intensité de connaissances dans le cadre du *benchmark* d'évaluation KILT montrent le potentiel de notre approche.

ABSTRACT

Enhancing Pre-trained Language Models via Mutual Knowledge Distillation

Knowledge bases are key resources in a wide range of knowledge intensive applications. However, their incompleteness inherently limits their use and gives rise to the importance of their completion. To this end, an open-world view has recently been held in the literature by coupling the ability of knowledge bases to represent factual knowledge, with the abilities of pre-trained language models (PLMs) to capture high-level and contextual linguistic knowledge from large-scale text corpora. In this work, we propose a distillation framework for knowledge base completion where PLMs leverage soft labels in the form of entity and relations predictions provided by a knowledge base embedding model, while keeping their power of entity prediction over large-scale of texts. To better fit with the task of knowledge completion, we extend the traditional masked language modelling of PLMs toward predicting entities and related entities in context. Experiments using the knledge intensive tasks within the standard KILT evaluation benchmark shows the potential of our proposed approach.

MOTS-CLÉS : Complétion de graphe de connaissances, enrichissement des modèles de langue pré-entraînés, distillation des connaissances.

KEYWORDS: Knowledge completion, Enhanced Pre-trained Language Models, Knowledge Distillation.

Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA), pages 139–156.

1 Introduction

Une base de connaissances (KB) est un graphe multirelationnel comprenant des entités et des relations et contenant des faits sous la forme de triplets *(entité tête (h), relation (r), entité queue (t))*. Les alignements entre le langage naturel et les triplets de la base de connaissances (KB) sont des éléments essentiels d'un large éventail de tâches de traitement du langage naturel (TAL) telles que l'extraction de relations (RE), la complétion de la base de connaissances (KBC) et la réponse aux questions (QA). Comme exemple récent dans cette direction de recherche, le *leaderboard* KILT (Petroni *et al.*, 2021) a popularisé deux collections à forte intensité de connaissances, T-REx (Elsahar *et al.*, 2018) et zsRE (Levy *et al.*, 2017), et fournit une collection Wikipédia alignés en tant que source de connaissances externe. En particulier, l'objectif général de KBC (Bordes *et al.*, 2013; Betz *et al.*, 2022) consiste à combler les lacunes dans la connaissance actuelle d'une KB, en se basant sur les informations structurées sur les entités et les relations entre les entités. Plus formellement, la tâche consiste à calculer le score de plausibilité $f(h, r, t)$ de triplets (h, r, t) non présents dans la base de données, en se basant sur la connaissance capturée dans une ressource. Dans la littérature, les pipelines KBC sont généralement composés de plusieurs modules de base, y compris la classification des faits des entités, la mise en relation des entités, la prédiction des liens et les méthodes de classification des relations (Ellis *et al.*, 2015).

Une approche fréquemment adoptée dans la littérature pour KBC, s'appuie sur les plongements de graphes pour apprendre des représentations d'entités et des relations entre entités, notamment TransE (Bordes *et al.*, 2013) et TransH (Wang *et al.*, 2014). Cependant, tout en ayant permis de réaliser des progrès significatifs dans le domaine de la recherche sur le KBC, ces modèles suivent l'hypothèse du monde fermé en vertu de laquelle de nouvelles relations et de nouveaux types d'entités ne pourraient pas être découverts, ce qui nuit à leur capacité de généralisation à l'extérieur de la KB, et limite leur adéquation aux KB hautement évolutives (Shi & Weninger, 2018). Ainsi, une tendance de recherche émergent assouplit cette hypothèse en plaidant pour une interprétation du monde ouvert (Shi & Weninger, 2018) où les modèles sont capables de prédire soit des relations non vues, soit des entités non vues. Une façon intuitive d'aborder cette question est l'utilisation d'une ressource de connaissances externe qui fournit des idées sur les nouvelles entités et relations. Une première ligne de travail tente de tirer parti de ces connaissances supplémentaires dans une variété de tâches à forte intensité de connaissances, y compris, mais sans s'y limiter, le KBC. Un important corpus de travaux qui a attiré beaucoup d'attention, exploite en particulier la grande capacité des modèles de langage pré-entraînés (PLM) tels que BERT (Devlin *et al.*, 2019), qui peut modéliser des relations sémantiques complexes qui peuvent être observées dans le langage du monde ouvert (Lewis *et al.*, 2020; Guu *et al.*, 2020). En conséquence, de nombreux modèles ont été développés, soit en incorporant les plongements de la KB comme caractéristiques d'entrée pour les PLMs (Bordes *et al.*, 2013; Lin *et al.*, 2015), soit en apprenant à représenter les entités directement à l'intérieur du modèle de langage grâce à un objectif de pré-entraînement guidé par les plongements de la KB (Poerner *et al.*, 2020; Yamada *et al.*, 2020; Wang *et al.*, 2021). Ces modèles se sont révélés être des alternatives appropriées pour améliorer les tâches à forte intensité de connaissances (par exemple, la complétion de *slots* [1]), mais sans que leur impact sur le KBC ne soit clairement établi (Yang *et al.*, 2021). Une autre ligne de travail cible spécifiquement le KBC en utilisant des sources textuelles dans le cadre d'apprentissage sous la forme de descriptions d'entités au niveau local (Han *et al.*, 2018; Shi & Weninger, 2018; Oh *et al.*, 2022) ou de statistiques de corpus au niveau global (Yao *et al.*, 2019; Chen *et al.*, 2019). Hormis le modèle d'intégration de graphes régularisés assisté par le texte présenté dans (Chen *et al.*, 2019), la

1. *Slot filling* en anglais.

principale caractéristique commune de ces travaux est qu'ils effectuent un alignement conjoint des espaces sémantiques, ce qui soulève des problèmes critiques dans le cas d'espaces hétérogènes avec des entités qui ne se chevauchent pas. D'un point de vue radicalement différent de tous les travaux cités ci-dessus, y compris (Chen *et al.*, 2019), nous proposons un PLM piloté par la KB en laissant les représentations d'entités de la KB et du PLM apprises dans leurs espaces inhérents mais partageant des étiquettes souples pendant une étape additionnelle de pré-apprentissage via la distillation des connaissances. Notre idée sous-jacente est d'élargir les capacités de prédiction d'un PLM en injectant des connaissances relationnelles à partir de la KB.

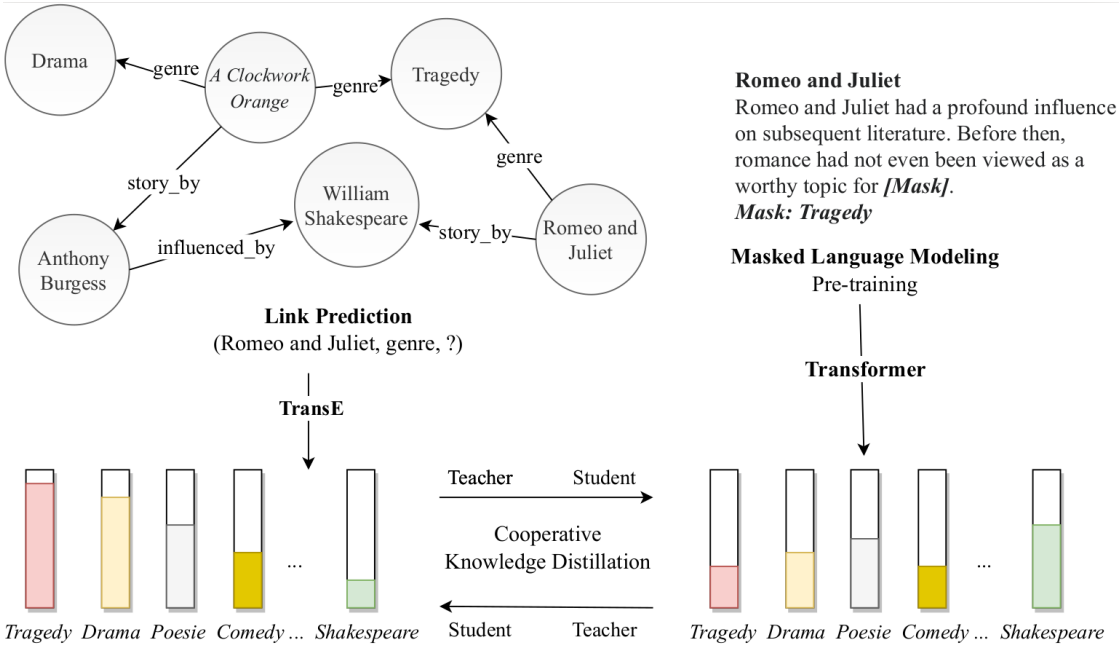


FIGURE 1 – Distillation de la connaissance entre le PLM et KB pour la tâche de complétion de connaissances.

Un exemple de cette distillation entre les PLM et les KB est illustré à la figure 1, où une base de connaissances de très petite taille (6 entités, 5 relations et 6 triplets) est utilisée pour entraîner un modèle de plongement de la KB sur la tâche de prédiction de liens, afin d'extraire les principales entités susceptibles de remplir le triplet *[Romeo and Juliet, genre, ?]*. De même, un texte est utilisé pour entraîner le PLM en utilisant la stratégie de masquage où la réponse est la même que pour la tâche de prédiction de liens. Dans une configuration traditionnelle, les modèles à source unique basés sur les PLM sont entraînés à prédire un *token* masqué tel que *"tragedy"* dans la phrase *"Romeo and Juliet is a [MASK]"*. Nous proposons plutôt d'apprendre à notre PLM à récupérer non seulement le *token* masqué de la vérité de terrain, mais aussi à produire des *logits* plus élevés pour les *tokens* qui y sont liés, c'est-à-dire *"drame", "poesie", .., "comedy"* , où ces étiquettes souples de haute qualité sont obtenues à partir d'un modèle plongement de la KB, tel que TransE (Bordes *et al.*, 2013).

Pour enseigner les représentations d'entités et de relations PLM, nous concevons une stratégie de masquage au niveau de l'entité qui force la modélisation traditionnelle du langage masqué (MLM) à se concentrer sur les entités mentionnées dans un corpus. En outre, pour enrichir les MLM avec des connaissances factuelles dans la KB, nous étudions la définition de deux variantes de fonctions de perte de distillation. Dans la première variante, nous considérons une distillation traditionnelle professeur-élève où le modèle PLM tire parti des prédictions du modèle de plongements de la KB. Dans la seconde variante, nous considérons une distillation coopérative, telle qu'elle a été explorée

précédemment, exclusivement pour les plongements des KBs (Sourty *et al.*, 2020), où le modèle PLM tire parti d'un modèle de plongements de la KB distillé à son tour.

Les principales contributions de notre article sont les suivantes :

— Un nouveau PLM basé sur des connaissances pour des tâches à forte intensité de connaissances, s'appuyant sur des stratégies de distillation axé sur les prédictions d'entités entre le PLM d'une part et la prédiction de liens du modèle d'intégration de la KB d'autre part.
— Une évaluation approfondie des PLM standard par rapport à notre PLM enrichi sur deux tâches à forte intensité de connaissances, à savoir T-REx et zsRE, dans le cadre du *benchmark* KILT (Petroni *et al.*, 2021).

Le reste de cet article est structuré comme suit. La section 2 présente les travaux connexes. Dans la section 3, nous présentons notre procédure de pré-entraînement coopératif. Dans la section 4, nous présentons et discutons les résultats expérimentaux. Enfin, la section 5 conclut l'article.

2 Travaux connexes

Bien que de multiples travaux aient été proposés dans le contexte des modèles enrichis par des connaissances, à notre connaissance, aucune de ces méthodes ne repose sur l'*apprentissage coopératif* entre deux espaces distincts, l'un dédié à la représentation du langage et l'autre à la représentation des connaissances. Nous présentons ici les avancées récentes sur ces trois sujets.

2.1 PLMs et PLMs enrichis avec des connaissances

Les modèles de représentation du langage naturel tels que BERT (Devlin *et al.*, 2019) peuvent modéliser des relations sémantiques complexes qui peuvent être observées dans le langage.

Cependant, les informations contenues dans les bases de connaissances peuvent être intégrées à ces modèles par un processus d'apprentissage supplémentaire. Il y a principalement deux approches d'intégration des connaissances issues des KB dans les PLMs : 1) incorporer les plongements de la KB comme caractéristiques d'entrée pour les PLM et se fier à leur capacité à à représenter la structure particulière des KB à l'aide d'opérations de décomposition tensorielle (Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015; Sun *et al.*, 2019) ; 2) apprendre à représenter les entités directement dans le modèle de langue (Poerner *et al.*, 2020) en injectant des connaissances dans BERT et en alignant les plongements d'entités avec les vecteurs de morceaux de mots sur la base d'une transformation linéaire. Par exemple, les travaux de Zhang *et al.* (2019) s'appuient sur des plongements d'entités incorporées via un mécanisme d'agrégation intégré directement dans l'architecture PLM. Peters *et al.* (2019) introduisent, quant à eux, le mécanisme d'attention pour incorporer les connaissances factuelles des synsets de Wordnet et définissent une fonction objectif de pré-entraînement pour la tâche de liaison référentielle d'entités. Dans Yamada *et al.* (2020), les auteurs proposent un mécanisme d'auto-attention sensible aux entités en dédiant les paramètres de la matrice de requête aux entités, de manière similaire à Wang *et al.* (2021), mais ce dernier utilise la prédiction de liens comme objectif complémentaire à la MLM et s'appuie sur les descriptions textuelles des entités pour apprendre les représentations des triplets de la KB.

2.2 Distillation des connaissances

Le processus de distillation des connaissances (*Knowledge Distillation KD*) a été largement utilisé comme une méthode compétitive pour transférer les connaissances d'un modèle large qualifié de professeur (*Teacher*) à un autre modèle moins large qualifié de modèle élève (*Student*) selon le principe de compression de modèles. Bucila *et al.* (2006) ont utilisé ce mécanisme pour compresser la taille de plusieurs modèles jouant le rôle de professeurs en un seul, où un modèle léger joue le rôle d'un élève. Dans Romero *et al.* (2015); Yim *et al.* (2017), les auteurs ont distillé les représentations internes du professeur pour accroître la capacité de l'élève à généraliser ses prédictions. Des travaux récents ont aussi adapté le concept de KD aux tâches de compréhension du langage naturel. Dans Saleh *et al.* (2020), les auteurs ont amélioré la traduction automatique neuronale à faibles ressources par une approche d'apprentissage par transfert de plusieurs modèles vers un seul modèle. Lai *et al.* (2020) ont adapté la procédure d'auto-distillation pour générer des pseudo-étiquettes sur la tâche d'extraction de phrases-clés. Alors que la plupart des méthodes attribuent un rôle unique aux modèles, soit le professeur ou l'élève, des travaux récents (Zhang *et al.*, 2018; Sourty *et al.*, 2020; Guo *et al.*, 2020; Sun *et al.*, 2021) ont proposé une approche d'apprentissage coopératif en faisant jouer aux modèles de façon alternée les rôles de professeur et d'élève et montrent que cela est bénéfique pour l'ensemble.

3 PLM enrichi via la distillation coopérative des connaissances

Considérons deux sources d'informations :

— une KB comme un graphe $(\mathcal{E}, \mathcal{R})$ composé d'entités $\mathcal{E} = \{e_1, \ldots, e_{N_e}\}$, un ensemble de relations $\mathcal{R} = \{r_1, \ldots, r_{N_r}\}$, et un ensemble de triplets positifs, ou faits, (e_x, r_w, e_y) noté T^+ parmi tous ceux possibles dans $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$.

— une collection de textes sous la forme d'une séquence de *tokens* $(t_1, t_2, ..., t_{N_t})$, où certains des *tokens* font référence à des entités, par exemple $t_i = e_j$.

Du point de vue des KB, la tâche de complétion de connaissances peut être définie comme suit : étant donné une requête composée d'une entité $e_i \in \mathcal{E}$ et d'une relation $r_k \in \mathcal{R}$, l'objectif de la tâche consiste à retrouver une entité $e_j \in \mathcal{E}$ qui permet de reconstruire le triplet positif (e_i, r_k, e_j). Notre modèle enrichi étend la cible de reconstruction basée sur une collection de textes. Considérons le tuple (e_j, S_{e_j}) où S_{e_j} est l'ensemble des phrases où e_j participe au moins une fois, par exemple $S_{e_j} = \{(t_{l_1}, t_{l_1+1}, ..., e_j, ...), (t_{l_2}, t_{l_2+1}, ..., e_j, ...), ...\}$. Ainsi, *du texte à la KB*, au lieu de définir la cible de la requête (e_i, s_k) avec une seule réponse correcte (e_j), nous proposons d'étendre la liste des réponses candidates avec $\{e_j^0, e_j^1, .., e_j^n\}$ où toutes ces entités sont obtenues en masquant l'entité e_j sur S_{e_j} [2]. De même, *de la KB au texte*, nous proposons pour chaque séquence dans S_{e_j} d'étendre la réponse correcte lors du masquage de e_j à la liste complète des candidats obtenus comme réponse à la requête (e_i, r_k). Notez que la reconstruction est possible dans les deux sens si un modèle est capable d'utiliser une requête ((e_i, r_k) ou un élément de S_{e_j} avec e_j masqué) en entrée et de produire un ensemble d'entités candidates avec une probabilité. Ainsi, l'utilisation de n'importe quelle combinaison entre KB et/ou documents textuels pourrait être considérée dans le cadre de la stratégie proposée, où l'entité cible e_j peut avoir une probabilité d'apparition maximale suivie de la liste de candidats supplémentaires.

2. Pour des raisons pratiques, un sous-ensemble de S_{e_j} sélectionné aléatoirement est utilisé à chaque itération.

L'objectif de notre modèle est double : 1) améliorer la représentation de la requête par le modèle du professeur au travers l'ensemble des représentations des entités $\{e_i^0, e_i^1, ..., e_i^n\}$ qui sont susceptibles de remplacer l'entité e_i sur la base de son voisinage dans l'espace de représentation de la KB ; 2) augmenter la capacité du modèle à proposer des candidats pertinents en apprenant un ensemble de représentations d'entités qui peuvent être utilisées comme substituts de l'ensemble des réponses attendues $\{e_{j_0}^0, e_{j_1}^0, .., e_{j_m}^0\}, .., \{e_{j_0}^n, e_{j_1}^n, .., e_{j_m}^n\}$ où $\{e_{j_0}^0, e_{j_1}^0, .., e_{j_m}^0\}$ est l'ensemble des entités qui peuvent remplacer la réponse cible e_j^0.

Nous soutenons globalement l'idée que la mise à jour simultanée des modèles de KB et des PLM permet de construire un espace où la distillation des connaissances est plus facile à opérer.

3.1 Transférer les probabilités des entités

Bien qu'il existe des ressources d'entités et de contenus alignés, la complexité de cette tâche peut être augmentée par les particularités des modèles PLM tels que l'utilisation de morceaux de mots. Ainsi, nous avons aligné la tâche de prédiction de liens et la tâche MLM pour transférer les connaissances encodées par chaque modèle. Lors de l'exécution de la tâche MLM, nous masquons dans 30% des cas une entité afin d'entraîner notre modèle via la fonction objectif de la distillation (Cf. Section 3.2), et dans 70% des cas, nous appliquons la procédure standard de MLM définie par Devlin *et al.* (2019). Le modèle doit récupérer le *token* original avec une fonction objectif basée sur l'entropie croisée lorsqu'il s'agit de la procédure standard. Nous avons conservé les deux fonctions objectifs afin que notre modèle bénéficie de la KD sur les entités tout en maintenant sa capacité de prédiction sur le vocabulaire courant et en évitant de dégrader les connaissances acquises lors de l'apprentissage initial du modèle.

Afin d' *estimer des probabilités d'entités à partir d'un PLM*, nous calculons d'abord des probabilités à partir d'un PLM basé sur des phrases composées d'une entité (e_i) et d'un contexte ($S_{e_i}^k$, la séquence k dans S_{e_i}), où l'entité e_i est masquée.

Ensuite, nous estimons la probabilité de toute entité $e_l \in \mathcal{E}$ d'être pertinente pour le contexte donné $S_{e_i}^k$ comme suit :

$$\hat{\mathcal{P}}(e_l|S_{e_i}^k, \theta_{mlm}) = \frac{\exp(mlm(e_l, S_{e_i}^k))}{\sum_{e_j \in \mathcal{E}} \exp(mlm(e_j, S_{e_i}^k))} \quad (1)$$

où la fonction mlm est notre prédicteur PLM pour la tâche MLM et θ_{mlm} sont ses paramètres. Notez qu'idéalement, le prédicteur donnera une probabilité maximale à l'entité masquée, par exemple e_i. Comme inconvénient, nous pouvons souligner que le vocabulaire d'un PLM est un nombre limité de séquences de caractères se répétant fréquemment. Par conséquent, une entité $e_i \in \mathcal{E}$ peut être composée de m_i morceaux de mots. Pour résoudre ce problème, nous avons sélectionné comme étiquette une mention de l'entité e_i qui fait déjà partie du vocabulaire du PLM et alternativement la mention la plus fréquente, c'est-à-dire que l'entité *"Rio de Janeiro"* devient *"Rio"* si cette dernière est sa mention la plus fréquente [3]. Lorsqu'aucune mention n'a été trouvée dans le vocabulaire, nous avons ajouté l'ensemble des entités $\mathcal{E}$ au vocabulaire du PLM en initialisant chaque entité ajoutée

3. Un étude détaillé sur l'impact de cette stratégie n'est pas abordé dans cet article, cependant, dans notre contexte, son utilisation est indispensable pour simplifier le processus de distillation.

comme la moyenne de ses *tokens* incorporés et en mettant à jour la dernière couche en conséquence avec le nombre mis à jour de *tokens* cibles comme suit :

$$embedding(e_i) = \frac{\sum_{m_j \in tokenizer(e_i)} embedding(m_j)}{|tokenizer(e_i)|} \tag{2}$$

De façon duale, afin d' *estimer les probabilités d'entités à partir d'un modèle d'injection de KB*, nous nous appuyons sur les modèles de représentations de KB par des plongements (*embedding*), telles que TransE (Bordes *et al.*, 2013), pour apprendre les représentations des entités tout en tenant compte de sa structure particulière en tant qu'ensemble d'entités connectées par des relations. Ensuite, nous calculons les probabilités de chaque entité par rapport à la relation et à une entité queue/tête (*tail*, *head*) en utilisant :

$$\hat{\mathcal{P}}(e_i|e_j, r_k, \theta_{lp}) = \frac{\exp(f(e_i, r_k, e_j))}{\sum_{e_{i\prime} \in \mathcal{E}} \exp(f(e_{i\prime}, r_k, e_j))} \tag{3}$$

où $f(.,.,.)$ est un modèle de prédiction de liens tel que TransE (Bordes *et al.*, 2013) et θ_{lp} ses paramètres, et l'entité e_j et la relation r_k sont obtenues de la KB si le triplet existe, par exemple $(e_i, r_k, e_j) \in T^+$. Notez que la position de tête ou de queue de e_i dans le triplet n'affecte que l'ordre des premier et troisième paramètres dans $f(.,.,.)$.

3.2 Fonction objectif coopérative

Notre procédure coopérative implique la mise à jour successive des paramètres du PLM et du plongement de la KB qui jouent alternativement les rôles de modèles de professeur et de l'élève, comme suggéré dans des travaux antérieurs (Zhang *et al.*, 2018; Sourty *et al.*, 2020; Guo *et al.*, 2020). Nous formulons l'objectif d'apprentissage mutuel entre les tâches de prédiction de liens et de MLM comme suit :

$$\mathcal{L}^{kd} = \mathcal{D}(\hat{\mathcal{P}}(e_j|e_i, r_k, \theta_{lp}), \hat{\mathcal{P}}(e_j|S^k_{e_i}, \theta_{mlm})) \tag{4}$$

où θ_{lp} et θ_{mlm} sont les paramètres d'un modèle pour la KB et d'un PLM, respectivement. e_i est une entité $\in \mathcal{E}$ qui est mentionnée et masquée dans le contexte c_i. Comme dans des travaux récents sur la distillation (Hinton *et al.*, 2015; Micaelli & Storkey, 2019), nous utilisons la fonction de divergence de Kullback-Leibler (KL) comme mesure de distance $\mathcal{D}$. Nous adaptons la divergence de KL, et par conséquence $\mathcal{L}^{kd}$, pour distiller la connaissance du modèle professeur vers l'étudiant en conséquence du rôle que chaque modèle prend dans une itération.

Afin de combiner efficacement les fonctions objectifs de prédiction de lien dans la KB et MLM du PLM en vue de stabiliser la convergence de l'apprentissage coopératif, nous avons appliqué la normalisation proposée dans Zoph *et al.* (2020) pour formuler l'objectif global. La fonction objectif de notre PLM enrichi est alors :

$$\mathcal{L}_{plm} = \frac{1}{1 + \alpha_{mlm}} \left(\mathcal{L}^{kd} + \alpha_{mlm} \frac{\overline{\mathcal{L}^{kd}}}{\overline{\mathcal{L}_{mlm}}} \mathcal{L}_{mlm} \right) \tag{5}$$

où $\overline{\mathcal{L}^{kd}}$ et $\overline{\mathcal{L}_{mlm}}$ désignent les moyennes pondérées exponentielles des objectifs de distillation des connaissances et de modélisation du langage masqué, respectivement.

De même, la fonction objectif de notre modèle de KB enrichi est :

$$\mathcal{L}_{kb} = \frac{1}{1+\alpha_{lp}} \left(\mathcal{L}^{kd} + \alpha_{lp} \frac{\overline{\mathcal{L}^{kd}}}{\overline{\mathcal{L}_{lp}}} \mathcal{L}_{lp} \right) \tag{6}$$

où $\overline{\mathcal{L}_{lp}}$ désigne les moyennes pondérées exponentielles des objectifs de prédiction de lien.

4 Évaluation expérimentale et résultats

4.1 Configurations et modèles de référence

La configuration de notre proposition de PLM enrichi est désignée par `Cooptiv`. Dans cette configuration, les deux modèles PLM et de représentation de KB sont mis à jour via la distillation coopérative des connaissances et sur la base de leurs tâches respectives, c'est-à-dire MLM suivant la fonction objectif dans l'équation 5 et sur la prédiction de liens suivant la fonction objectif dans l'équation 6. Nous comparons notre modèle à deux [4] configurations de référence distinctes qui sont :

— `Vanilla` : Les deux modèles sont entraînés sur leurs tâches respectives, c'est-à-dire la MLM et la prédiction de liens. La distillation des connaissances n'est pas utilisée dans cette stratégie.
— `Knowldg` : Les deux modèles sont entraînés sur leurs tâches respectives, c'est-à-dire la MLM et la prédiction de liens. Seul le MLM bénéficie de la distillation via la fonction objectif dans l'équation 5.

Ces configurations partagent les mêmes hyperparamètres et ont été entraînées à l'aide de deux PLM distincts, DistillBERT (Sanh *et al.*, 2019) un modèle à base d'un *transformer* de 44 millions de paramètres, noté `PLM-A`, et BERT-base un autre modèle à base d'un *transformer* de 110 millions de paramètres (Devlin *et al.*, 2019), noté `PLM-B`.

Nous avons entraîné tous les modèles avec l'optimiseur Adam, avec un taux d'apprentissage de 5e-8 et une taille de lot de 32. En ce qui concerne le modèle de plongement de la KB, nous avons utilisé le modèle standard TransE avec une dimension de plongement de 500 pour les relations et les entités, un taux d'apprentissage de 5e-6, Adam comme optimiseur et une taille de lot de 512. Pour chaque triplet positif, nous avons généré 512 triplets corrompus suivant la fonction objectif pour la prédiction de liens adverses définie par (Sun *et al.*, 2019) avec un paramètre de marge γ fixé à 6. De plus, nous suivons (Zoph *et al.*, 2020) pour définir le taux de décroissance de la moyenne mobile exponentielle des fonctions objectifs (égale à 0,9997) dans les équations 5 et 6. Enfin, nous avons fixé les paramètres dédiés à la normalisation des fonctions objectifs, α_{lp} et α_{mlm}, égale à 0,5.

4. Une troisième configuration, de texte à la KB uniquement, a été ignorée car le PLM résultat est, dans ce cas, équivalent à `Vanilla`.

4.2 Évaluation intrinsèque

4.3 Jeu de données, pré-traitements et métriques

Nous avons utilisé le jeu de données standard FB15K-237 comme KB principale et les métriques d'évaluation standards, notamment HITS@K et MRR. Les statistiques de ce jeu de données sont présentées dans le Tableau 1 (colonne de gauche). Cependant, comme un corpus de texte est nécessaire pour la tâche MLM, nous avons aligné les entités FB15K-237 et leurs mentions dans Wikipédia en utilisant des hyperliens pour effectuer conjointement les tâches MLM et de prédiction de liens. Nous avons échantillonné 8 millions de phrases de Wikipédia qui mentionnent au moins une entité de la partition d'entraînement FB15K-237. Les statistiques des corpus de textes sont présentées dans le Tableau 1 (colonne de droite). Notre échantillon de la Wikipédia présente un taux de couverture significatif des entités FB15K-237 avec au moins une mention de 86,1% des entités et 74,9% des triplets (ensembles d'entraînement, de validation et de test combinés). Nous avons également échantillonné 60 000 phrases conservées afin de construire un ensemble de validation et un ensemble de test pour l'évaluation intrinsèque. Pour assurer la couverture des entités utilisées composées de plusieurs mots, nous avons ajouté 12 230 mentions manquantes au vocabulaire, comme décrit dans la section 3.1. Les entités restantes étaient présentes dans les PLMs utilisés. Pour mesurer la qualité du PLM appris, nous avons utilisé la mesure de perplexité standard (*PPL*). Notez que comme l'information sur le *token* est connue, nous pouvons calculer la perplexité en considérant si le *token* attendu est une entité ou non. Ainsi, nous avons calculé la métrique "*PPL Entités*" en mesurant la perplexité exclusivement sur les mentions des entités de notre KB dans Wikipédia.

Jeu de données	**FB15K-237**	**Wikipédia**
# Entités	14541	12516
# Relations	237	-
# Entraînement	272115	8000000
# Validation	17535	30000
# Test	20466	30000
Couverture des entités de la KB	-	86.1%
Couverture des triplets de la KB	-	74.9%

TABLE 1 – Statistiques de la KB FB15K-237 et des corpus de textes dédiés aux tâches de prédiction de liens et de MLM, respectivement. Les taux de couverture du corpus textuel par rapport à la KB sont fournie en termes d'entités et de triplets.

4.4 Résultats et discussion

Les résultats des trois configurations utilisant les deux PLM sont présentés dans le Tableau 2. Sans surprise, comme les `PLM-B` ont plus du double de paramètres que les `PLM-A`, les modèles `PLM-B` surpassent clairement les `PLM-A` à la fois selon la perplexité (PPL) et selon la perplexité sur les entités (PPL Entités). De même, comme on pouvait s'y attendre, les configurations enrichies de connaissances (`Knowldg` et `Cooptiv`) ont des performances qui dépassent celles des modèles `Vanilla` homologues en termes de la mesure *PPL Entités*. Cela indique que les deux PLM ont été capables de capturer les signaux d'entité fournis par les injections de connaissances issues de la KB.

Tâche	Modélisation du Langage Masqué		Prédiction de lien				
PLM	**PPL Entités**	**PPL**	**plongement KB**	**HITS@1**	**HITS@3**	**HITS@10**	**MRR**
Vanilla PLM-A	10.12	7.55	TransE	**22.53**	**36.27**	**52.15**	**0.32**
Knowldg PLM-A	**8.36**	**7.37**	TransE	**22.53**	**36.27**	**52.15**	**0.32**
Cooptiv PLM-A	8.38	7.41	TransE	21.02	34.58	50.21	0.30
Vanilla PLM-B	7.81	**6.02**	TransE	**22.53**	**36.27**	**52.15**	**0.32**
Knowldg PLM-B	**7.28**	6.40	TransE	**22.53**	**36.27**	**52.15**	**0.32**
Cooptiv PLM-B	7.31	6.34	TransE	20.95	34.55	50.19	0.30

TABLE 2 – Évaluation intrinsèque des PLM standard et PLM enrichi et du modèle de plongement de la KB. Les meilleures valeurs pour chaque PLM sont indiquées en **gras**.

Plus précisément, le modèle `Cooptiv PLM-A` améliore la perplexité sur les entités par rapport au `Vanilla PLM-A` (8,38 contre 10,12), et le `Cooptiv PLM-B` obtient un score de 7,31 contre 7,81 pour le `Vanilla PLM-B`. Enfin, en ce qui concerne le `PLM-B`, les deux stratégies `Knowldg` et `Cooptiv` dégradent légèrement la mesure de perplexité : 6,40 et 6,34 contre 6,02 pour `Vanilla`. Bien que l'ordre ne soit pas similaire pour `PLM-B`, les différences sont faibles, ce qui suggère que l'impact la perplexité calculée sur les mots est faible également. Ainsi, dans l'ensemble, les stratégies `Cooptiv` et `Knowldg` préservent la capacité des PLM à traiter les *tokens* les plus fréquents.

Nous vérifions la précision de chaque modèle TransE via l'évaluation de la prédiction de liens et reportons les résultats dans le Tableau 2. Nous avons mesuré les scores de prédiction de liens de nos modèles de plongement de la KB en utilisant l'ensemble des triplets de test de FB15K-237. Notez que pour le modèle `Vanilla` et `Knowldg` les valeurs correspondent à un modèle TransE standard car sur ces configurations, il n'y a pas d'impact sur les plongements de la KB. Pour les deux PLMs, les résultats de TransE ne bénéficient pas de la distillation des connaissances mais ne conduisent pas non plus à des résultats aberrants : TransE en paire avec `Cooptiv PLM-A` ou avec `Cooptiv PLM-B` conduit à une diminution de la métrique HITS@3 de -4.7% dans les deux cas. TransE n'est pas excessivement biaisé en faveur du modèle de langue malgré le fait que nous ayons fixé le facteur de normalisation α_{lp} à $0,5$ (voir l'équation 6) et qu'il accorde de l'importance aux pseudo-étiquettes de `PLM-A` et `PLM-B`. Nous pensons que les natures différentes et les objectifs distincts entre les PLMs et les plongements de la KB font qu'il est plus difficile pour la stratégie coopérative d'obtenir des améliorations sur la tâche de prédiction de liens, mais qu'elle peut aider à un meilleur alignement entre les deux espaces. Le compromis entre la complexité de l'optimisation et la qualité des données de distillation (Stanton *et al.*, 2021) peut expliquer ce résultat, car un élève qui reproduit un professeur via la distillation des connaissances ne conduit pas systématiquement à une amélioration.

Soit de fréquence→	50			150			300		
Modèle↓Métrique→	**P@1**	**P@10**	**P@100**	**P@1**	**P@10**	**P@100**	**P@1**	**P@10**	**P@100**
Vanilla PLM-A	**2.06**	6.19	16.49	**2.12**	7.67	17.46	**2.82**	**10.06**	24.54
Knowldg PLM-A	1.03	**8.25**	**19.59**	1.32	7.67	19.84	2.45	10.43	26.38
Cooptiv PLM-A	1.03	**8.25**	**19.59**	1.32	7.67	**20.11**	2.70	9.94	**26.87**
Vanilla PLM-B	2.06	8.25	18.56	**2.65**	6.88	17.46	**2.82**	8.10	22.33
Knowldg PLM-B	**4.12**	**9.28**	**21.65**	1.85	7.94	20.63	2.33	9.45	25.89
Cooptiv PLM-B	3.09	**9.28**	**21.65**	1.59	**8.20**	**20.90**	2.21	**9.69**	**26.01**

TABLE 3 – Précision de MLM à k, avec k = {1, 10, 100}. Les seuils de 50, 150 et 300 indiquent la limite supérieure de fréquence de l'entité cible dans le corpus.

Pour mieux saisir l'amélioration de la perplexité sur les entités observée dans nos modèles enrichis, nous avons également mesuré la capacité d'un PLM à récupérer une entité masquée en fonction de sa fréquence d'apparition dans le corpus d'entraînement et avons reporté les résultats avec différents seuils de fréquence dans le Tableau 3. Les résultats montrent que moins une mention d'entité est fréquente, plus il sera difficile pour le modèle de langue de la retrouver. Dans la plupart des cas, la précision diminue lorsqu'un seuil de fréquence plus bas est utilisé pour un modèle donné. Cette évaluation reflète la difficulté des modèles de langue à s'adapter aux entités peu fréquentes ou aux nouveaux domaines. Les modèles `Vanilla PLM-A` et `Vanilla PLM-B` ne classent que 17,5% des entités masquées (avec un seuil < 150) dans les 100 premières entités. Les deux stratégies de distillation, `Cooptiv` et `Knowldg`, surpassent systématiquement la stratégie `Vanilla` pour les deux modèles PLM, `PLM-A` et `PLM-B`, en termes de P@100. De plus, la stratégie `Cooptiv` surpasse la stratégie `Knowldg` pour les valeurs de seuil de 150 et 300. Ces résultats suggèrent que les PLMs ont amélioré leur représentation interne de leurs entités via des pseudo-étiquettes sans avoir besoin de nombreux exemples explicites dans le corpus de textes.

4.5 Évaluation extrinsèque

4.5.1 Jeux de données et modèles de référence

Nous avons évalué tous les modèles sur deux jeux de données dédiés à des tâches orientées connaissances, T-REx (Elsahar *et al.*, 2018) et zsRE (Levy *et al.*, 2017) pour la complétion de slots (*Slot filling*). Le but de cette tâche consiste à récupérer tous les paramètres (*slots*), sous forme d'entités, qui composent une intention (question). Comme proposé dans (Petroni *et al.*, 2021), nous avons collecté 2 284 168 paires de questions et de réponses pour T-REx et 197 620 paires pour zsRE. Comme modèles de référence, nous avons opté pour BERT + DPR (Karpukhin *et al.*, 2020), BART + DPR, et RAG (Lewis *et al.*, 2020) fournis par le tableau de classement KILT. BERT + DPR est un *pipeline* initié par un système de recherche et un modèle extractif de réponses aux questions. La base BART + DPR est performante et bénéficie de son grand nombre de paramètres et de la capacité du lecteur à générer des réponses héritées des modèles de séquences à séquences. RAG est un *pipeline* de bout en bout affiné sur la tâche de complétion de slots basé sur un système de recherche appelé DPR et d'un lecteur BART. Enfin, DensePhrases^{10k} s'appuie sur le modèle de base SpanBERT (Joshi *et al.*, 2020).

4.5.2 Paramètres

Nous avons entraîné nos modèles avec un objectif d'extraction de réponses aux questions (QA). Nous avons aligné les questions telles que (e_i, s_k) et les passages de Wikipédia qui ont au moins une des réponses attendues pour la complétions des slots $\in \{e_j^0, e_j^1, .., e_j^n\}$. Nous avons entraîné les modèles sur T-REx pendant une seule époque avec l'optimiseur AdamW, avec un taux d'apprentissage fixé à 2e-5 et une taille de lot de 16. Sur zsRE, nous nous sommes appuyés sur cinq époques et l'optimiseur AdamW avec un taux d'apprentissage de 2e-5. Nous avons ajouté une régularisation à nos modèles sur les deux modèles de complétion de slots en fixant le coefficient de décroissance des poids d'AdamW à 0,01. Au moment de l'inférence, nous avons commencé par diviser en paragraphes la source de connaissances de 5,9 millions de documents partagée par KILT. Cela représente plus de 110 millions de paragraphes que nous avons indexés avec BM25. Ensuite, nous avons filtré les paragraphes les plus pertinents en suivant le cadre de recherche-lecture pour chaque requête. Nous avons retrouvé les

documents en utilisant le titre de la page Wikipédia et le contenu du paragraphe pour T-REx. Pour zsRE, nous avons utilisé uniquement le titre de la page Wikipédia, qui contient souvent les entités sujet. Nous avons sélectionné les champs utilisés par l'extracteur en évaluant l'ensemble du pipeline recherche-lecture sur le jeu de données de validation de T-REx et zsRE. Nous avons finalement extrait la réponse la plus probable parmi les 200 premiers paragraphes trouvés avec nos PLMs.

4.5.3 Métriques

Nous avons évalué nos modèles à l'aide du benchmark KILT. KILT évalue les performances d'un modèle sur 1) sa capacité à extraire des preuves (R-PREC, Recall@5), 2) la précision des candidats proposés par le système (Accuracy, F1), et 3) une combinaison des deux métriques de recherche et de précision (KILT-AC, KILT-F1). KILT-AC et KILT-F1 correspondant à une Accuracy et F1 pour lesquelles une réponse est correcte si le document qui a permis de la trouver est classé en premier. Par conséquent, les métriques KILT-AC <= Accuracy et KILT-F1 <= F1 sont utilisées car ils mettent l'accent sur l'interprétabilité.

	KILT-AC		**KILT-F1**		**R-Prec**		**Accuracy**		**F1**		**Recall@5**	
Model	T-REx	zsRE	T-REx	zsRE	T-REx	zsRE	T-REx	zsRE	T-REx	zsRE	T-REx	zsRE
Vanilla PLM-A	34.69	31.93	37.57	35.06	46.50	61.65	46.72	33.66	52.69	37.47	51.07	63.37
Knowldg PLM-A	34.96	32.23	37.77	35.15	46.90	59.68	46.36	34.65	51.86	38.18	50.89	62.04
Cooptiv PLM-A	**36.68**	34.13	**39.56**	37.22	**48.08**	61.33	49.04	36.22	54.61	40.33	**51.86**	63.85
Vanilla PLM-B	33.08	31.05	35.96	35.48	44.58	59.31	45.5	36.09	51.02	40.61	49.24	63.32
Knowldg PLM-B	32.18	28.79	35.01	32.54	43.94	57.20	44.44	32.64	50.77	37.43	49.20	60.53
Cooptiv PLM-B	34.38	35.32	37.34	39.55	46.56	**63.18**	46.42	38.54	51.88	44.03	50.38	**66.51**
BERT + DPR (*Petroniet al.*, 2021)	-	4.47	-	27.09	-	40.11	-	6.93	-	37.28	-	40.11
BART + DPR (*Petroniet al.*, 2021)	11.12	18.91	11.41	20.32	13.26	28.90	59.16	30.43	62.76	34.47	17.04	39.21
RAG (*Lewiset al.*, 2020; *Petroniet al.*, 2021)	23.12	36.83	23.94	39.91	28.68	53.73	**59.20**	44.74	**62.96**	49.95	33.04	59.52
DensePhrases 10^k (*Leeet al.*, 2021)	27.84	**41.34**	32.34	**46.79**	37.62	57.43	53.90	**47.42**	61.74	**54.75**	40.07	60.47

TABLE 4 – Performances en aval sur le jeu de données KILT. Nous présentons les résultats des trois stratégies distinctes, à savoir `Vanilla`, `Knowldg`, et `Cooptiv` pour `PLM-A` et `PLM-B`. La meilleure performance entre tous les modèles est indiquée en **gras**. L'existence d'une amélioration par rapport à l'homologue `Vanilla` est indiquée en colour vert.

4.5.4 Résultats et discussion

Le Tableau 4 résume les résultats de nos modèles sur la tâche de complétion de slots : `Vanilla`, `Knowldg`, et `Cooptiv` pour les deux `PLM-A` et `PLM-B`. Dans l'ensemble, notre modèle `Cooptiv PLM-A` atteint des performances compétitives sur le jeu de données T-REx par rapport à DensePhrases, avec une amélioration relative de 31,8% sur la métrique KILT-AC, 22,3% sur KILT-F1, 27,8% sur R-Prec et 29,4% sur Recall@5.

L'amélioration systématique des performances, en termes de R-Prec et Recall@5 par rapport à Petroni *et al.* (2021); Lee *et al.* (2021); Lewis *et al.* (2020), montrent que nos modèles de lecteurs se basent davantage sur des documents pertinents pour les stratégies `Vanilla, Knowledg, Cooptiv` en utilisant les deux modèles, `PLM-A` et `PLM-B`. Également, la stratégie `Cooptiv` surpasse systématiquement ses homologues `Vanilla` et `Knowldg` sur toutes les métriques (KILT-AC, KILT-F1, Accuracy, F1, et Recall@5), pour les deux `PLM-A` et `PLM-B`, démontrant l'intérêt de la distillation coopérative.

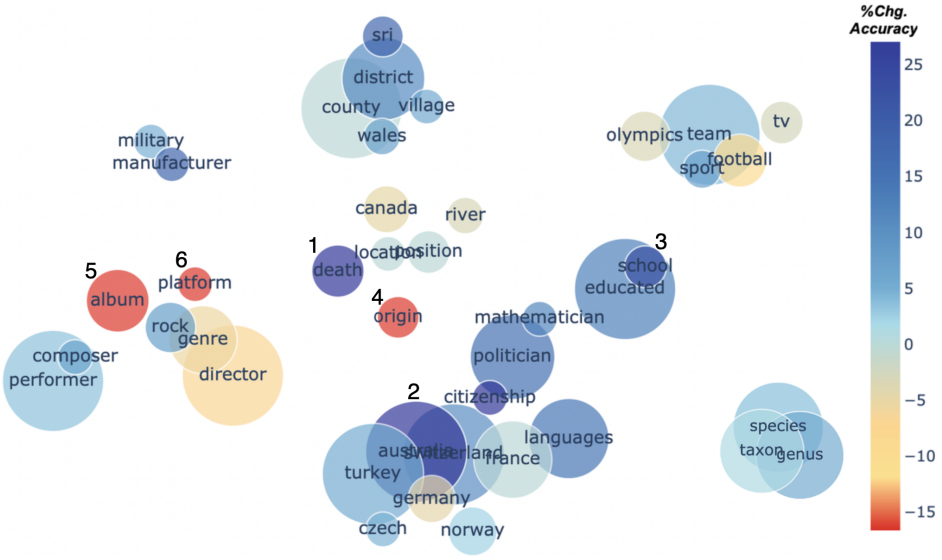


FIGURE 2 – Améliorations observées de la précision avec le modèle `Cooptiv PLM-A` sur le jeu de données T-REx par thème par rapport au modèle `Vanilla PLM-A`. La taille des classes est proportionnelle au nombre d'échantillons d'appartenance. Les 3 thèmes obtenant les meilleures performances ainsi que les 3 thèmes obtenant les plus basses performances sont numérotés de 1 à 6.

Pour mieux comprendre l'amélioration systématique observée sur le jeu de données T-REx, nous reportons dans la figure 2 l'amélioration relative de `Cooptiv PLM-A` par rapport à son homologue `Vanilla` en termes de précision. Nous avons construit les 41 thèmes en suivant la procédure définie par la bibliothèque Python BERTopic (Grootendorst, 2020) utilisant le PLM "all-MiniLM-L6-v2" Sentence (Reimers & Gurevych, 2019). BERTopic s'appuie sur un TF-IDF basé sur la classe pour extraire le *token* le plus représentatif comme descripteur de thème pour chaque regroupement. Nous pouvons constater que le `Cooptiv PLM-A` obtient de meilleurs résultats sur les thèmes *geography, science*, et *education* par rapport au `Vanilla PLM-A` avec une augmentation relative de la précision de 25% (indiqué par la couleur). Le modèle `Cooptiv PLM-A` améliore considérablement les résultats sur les groupes *1. death, 2. Australia*, et *3. school*, et réduit les performances sur les groupes *4. origin, 5. album*, et *6. platforme*. Les thèmes pour lesquels nous observons une amélioration se réfèrent à des entités sur-représentées dans les triplets d'entraînement de notre KB. 2931 triplets d'apprentissage de FB15K-237 référençant directement le thème *Australia* contre 66 référençant le sujet *platforme*. Les entités appartenant aux thèmes 1, 2 et 3 sont sur-représentées dans notre corpus Wikipédia. Par exemple, 1,4% des articles Wikipédia que nous avons utilisés pour améliorer `Cooptiv PLM` mentionnent une entité du thème *mort* contre 0,5% pour le thème *platforme*. 17,6% des *tokens* du thème *schools* récupérés par le TF-IDF basé sur la classe font partie des entités de FB15K-237 contre 2,6% pour le thème *origin*. Ainsi, un examen attentif des résultats de deux thèmes est présenté dans le Tableau 5. Ce tableau donne un aperçu des prédictions des modèles `Cooptiv PLM-A` et `Vanilla PLM-A`.

Pour les thèmes *Album* et *Australia* (voir Figure 2 et colonne Topique, Tableau 5), nous reportons les cinq meilleures réponses de chaque modèle pour des multiples requêtes de l'ensemble de données de validation T-REx. Nous distinguons les exemples pour lesquels notre modèle enrichi fournit la réponse attendue (indiquée en gras dans la colonne "Réponse") avec le type Q^+ (colonne Type) des exemples pour lesquels la version `Vanilla` est correcte avec le type Q^-. On peut ainsi constater que

pour le premier exemple, *[William Shakespeare [SEP] genre]*, Cooptiv retrouve la vérité terrain *drame de la renaissance anglaise* et propose avec succès l'entité FB15K-237 *tragedy*. Les entités géographiques sont sur-représentées (plus de 20%) dans les triplets de FB15K-237 et font référence à un *lieu de naissance* ou *mort* (thème *Death*), à la localisation d'un *University* (thème *school*), ou, plus globalement, à des infrastructures nationales (thème *Australia*). En effet, la distillation coopérative permet au modèle Cooptiv PLM-A de développer une meilleure compréhension des entités géopolitiques en répondant *united states* à la requête [*New York State Route 199 [SEP] country*] au lieu de lister les villes/régions comme son modèle homologue Vanilla.

Topique	Type	Requête	Modèle	Réponse
Album	Q^+	William Shakespeare [SEP] genre	Vanilla	shakespeare, sonneteers, comedies, dramatists, dramatist
			Cooptiv	**english renaissance**, tragedy, comedies, parodying, dramatist
		Phil Nimmons [SEP] occupation	Vanilla	architect, technologist, jazz musician, bandleaders, bullet
			Cooptiv	architect, technologist, **composer**, bandleaders, jazz musician
	Q^-	Sweet Memories [SEP] genre	Vanilla	**romance film**, romantic drama, country artist, country
			Cooptiv	country artist, adult contemporary, country tracks, willie nelson, country
		music manuscript [SEP] instance of	Vanilla	video game, **manuscript**, musical terminology, software, library
			Cooptiv	video game, library, terminology, musical terminology, software
Australia	Q^+	New York State Route 119 [SEP] country	Vanilla	utah, new york, nevada, washington, u.s. state of washington. state of utah
			Cooptiv	**united states**, utah, washington, new york, u.s. state of washington
		New York State Route 316 [SEP] country	Vanilla	georgia, ohio, new york, u.s. state of georgia, pickaway county
			Cooptiv	**united states**, georgia, ohio, new york, south bloomfield
	Q^-	Allied invasion of Sicily [SEP] country	Vanilla	malta, **italy**, greece, canada, kingdom of italy, tunisia
			Cooptiv	canada, malta, greece, sicily, kingdom of italy
		subregion of Finland [SEP] country	Vanilla	portugal, uganda, poland, colombia, **finland**
			Cooptiv	uganda, portugal, poland, united nations, colombia

TABLE 5 – Meilleures réponses sur le jeu de données T-REx récupérées par les versions Vanilla et Cooptiv de PLM-A classées par vraisemblance. Q^+ indique les requêtes où notre PLM amélioré est meilleur que son homologue vanille et vice versa pour Q^-. Les étiquettes correctes sont indiquées en **gras**.

5 Conclusion et travaux futurs

Dans cet article, nous avons proposé une approche basée sur la distillation pour enrichir un PLM sur des connaissances factuelles contenues dans une KB dans la perspective d'améliorer la connaissances du modèle. Nous avons proposé une stratégie de masquage axée sur les entités dans le but de permettre au PLM de capturer les relations implicites entre les entités en plus des relations entre les mots, comme c'est le cas dans les stratégies de masquage traditionnelles. Cette stratégie fait partie d'un cadre de distillation dans lequel le PLM utilise des étiquettes souples fournies par un modèle de plongement de la KB et vice-versa. L'évaluation expérimentale de deux tâches standard à forte intensité de connaissances, en utilisant T-REx et zsRE, a montré que nos PLM améliorés sont plus efficaces que leurs homologues vanille et sont compétitifs par rapport aux modèles de référence dans la plupart des métriques. Un examen plus approfondi des résultats du masquage a montré que nos PLMs améliorés comprennent mieux les représentations des entités qu'un PLM standard, mais qu'ils ont des difficultés pour les entités très peu fréquentes. En outre, la plupart des topiques de l'ensemble de données bénéficient de la représentation de notre modèle, avec une amélioration plus faible pour les topiques qui se rapportent davantage aux entités.

Nous prévoyons d'intégrer dans le cadre de la distillation la génération de pseudo-étiquettes pour les entités sous-représentées, en utilisant les étiquettes souples des entités voisines les plus proches fournies par la KB, à l'instar des approches proposées dans des travaux antérieurs (Tänzer *et al.*,

2022). L'évaluation expérimentale à grande échelle de l'utilisation des PLM proposés dans des tâches à forte intensité de connaissances, au-delà de la complétion de slots, mérite également d'être étudiée.

Références

BETZ P., MEILICKE C. & STUCKENSCHMIDT H. (2022). Supervised knowledge aggregation for knowledge graph completion. In *European Semantic Web Conference*, p. 74–92 : Springer.

BORDES A., USUNIER N., GARCIA-DURAN A., WESTON J. & YAKHNENKO O. (2013). Translating embeddings for modeling multi-relational data. In C. J. C. BURGES, L. BOTTOU, M. WELLING, Z. GHAHRAMANI & K. Q. WEINBERGER, Éds., *Advances in Neural Information Processing Systems 26*, p. 2787–2795 : Curran Associates, Inc.

BUCILA C., CARUANA R. & NICULESCU-MIZIL A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, p. 535–541, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/1150402.1150464.

CHEN T., ZHU S., WEN Y. & ZHENG Z. (2019). Knowledge graph completion with text-aided regularization. In *AAAI*.

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.

ELLIS J., GETMAN J., FORE D., KUSTER N., SONG Z., BIES A. & STRASSEL S. M. (2015). Overview of linguistic resources for the TAC KBP 2015 evaluations : Methodologies and results. In *Proceedings of the 2015 Text Analysis Conference, TAC 2015, Gaithersburg, Maryland, USA, November 16-17, 2015, 2015* : NIST.

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).

GROOTENDORST M. (2020). Bertopic : Leveraging bert and c-tf-idf to create easily interpretable topics. DOI : 10.5281/zenodo.4381785.

GUO Q., WANG X., WU Y., YU Z., LIANG D., HU X. & LUO P. (2020). Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

GUU K., LEE K., TUNG Z., PASUPAT P. & CHANG M. (2020). Retrieval augmented language model pre-training. 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. 3929–3938 : PMLR.

HAN X., LIU Z. & SUN M. (2018). Neural knowledge acquisition via mutual attention between knowledge graph and text. *AAAI*, **32**(1).

HINTON G., VINYALS O. & DEAN J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*.

JOSHI M., CHEN D., LIU Y., WELD D. S., ZETTLEMOYER L. & LEVY O. (2020). SpanBERT : Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, **8**, 64–77. DOI : 10.1162/tacl_a_00300.

KARPUKHIN V., OGUZ B., MIN S., LEWIS P., WU L., EDUNOV S., CHEN D. & YIH W.-T. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6769–6781, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.550.

LAI T., BUI T., KIM D. S. & TRAN Q. H. (2020). A joint learning approach based on self-distillation for keyphrase extraction from scientific documents. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 649–656, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.56.

LEE J., SUNG M., KANG J. & CHEN D. (2021). Learning dense representations of phrases at scale. In *Association for Computational Linguistics (ACL)*.

LEVY O., SEO M., CHOI E. & ZETTLEMOYER L. (2017). Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, p. 333–342, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/K17-1034.

LEWIS P., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., YIH W.-T., ROCKTÄSCHEL T., RIEDEL S. & KIELA D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems*, volume 33, p. 9459–9474 : Curran Associates, Inc.

LIN Y., LIU Z., SUN M., LIU Y. & ZHU X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, p. 2181–2187 : AAAI Press.

MICAELLI P. & STORKEY A. J. (2019). Zero-shot knowledge transfer via adversarial belief matching. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ-BUC, E. FOX & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 32 : Curran Associates, Inc.

OH B., SEO S., HWANG J., LEE D. & LEE K.-H. (2022). Open-world knowledge graph completion for unseen entities and relations via attentive feature aggregation. *Information Sciences*, **586**, 468–484. DOI : https ://doi.org/10.1016/j.ins.2021.11.085.

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.

PETRONI F., PIKTUS A., FAN A., LEWIS P., YAZDANI M., DE CAO N., THORNE J., JERNITE Y., KARPUKHIN V., MAILLARD J., PLACHOURAS V., ROCKTÄSCHEL T. & RIEDEL S. (2021). KILT : a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 2523–2544, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.200.

POERNER N., WALTINGER U. & SCHÜTZE H. (2020). E-BERT : Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics : EMNLP 2020*,

p. 803–818, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.71.

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* : Association for Computational Linguistics.

ROMERO A., BALLAS N., KAHOU S. E., CHASSANG A., GATTA C. & BENGIO Y. (2015). Fitnets : Hints for thin deep nets. *International Conference on Learning Representations*.

SALEH F., BUNTINE W. & HAFFARI G. (2020). Collective wisdom : Improving low-resource neural machine translation using adaptive knowledge distillation. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 3413–3421, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.302.

SANH V., DEBUT L., CHAUMOND J. & WOLF T. (2019). Distilbert, a distilled version of BERT : smaller, faster, cheaper and lighter. *CoRR*, **abs/1910.01108**.

SHI B. & WENINGER T. (2018). Open-world knowledge graph completion.

SOURTY R., MORENO J. G., SERVANT F.-P. & TAMINE-LECHANI L. (2020). Knowledge base embedding by cooperative knowledge distillation. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 5579–5590, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.489.

STANTON S., IZMAILOV P., KIRICHENKO P., ALEMI A. A. & WILSON A. G. (2021). Does knowledge distillation really work ? *Advances in Neural Information Processing Systems*, **34**.

SUN L., GOU J., YU B., DU L. & TAO D. (2021). Collaborative teacher-student learning via multiple knowledge transfer. *CoRR*, **abs/2101.08471**.

SUN Z., DENG Z.-H., NIE J.-Y. & TANG J. (2019). Rotate : Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

TÄNZER M., RUDER S. & REI M. (2022). Memorisation versus generalisation in pre-trained language models. In *ACL*, p. 7564–7578. DOI : 10.18653/v1/2022.acl-long.521.

WANG X., GAO T., ZHU Z., ZHANG Z., LIU Z., LI J. & TANG J. (2021). Kepler : A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, **9**, 176–194. DOI : 10.1162/tacl_a_00360.

WANG Z., ZHANG J., FENG J. & CHEN Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, p. 1112–1119 : AAAI Press.

YAMADA I., ASAI A., SHINDO H., TAKEDA H. & MATSUMOTO Y. (2020). LUKE : Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6442–6454, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.523.

YANG J., XIAO G., SHEN Y., JIANG W., HU X., ZHANG Y. & PENG J. (2021). A survey of knowledge enhanced pre-trained models. *ArXiv*, **abs/2110.00269**.

YAO L., MAO C. & LUO Y. (2019). KG-BERT : BERT for knowledge graph completion. *CoRR*, **abs/1909.03193**.

YIM J., JOO D., BAE J. & KIM J. (2017). A gift from knowledge distillation : Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

ZHANG Y., XIANG T., HOSPEDALES T. M. & LU H. (2018). Deep mutual learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 4320–4328.

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, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1139.

ZOPH B., GHIASI G., LIN T.-Y., CUI Y., LIU H., CUBUK E. D. & LE Q. V. (2020). Rethinking pre-training and self-training.

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

Constitution de sous-fils de conversations d'emails

Lionel Tadonfouet Tadjou[1, 2, 3] Eric De La Clergerie[1]
Fabrice Bourge[2] Tiphaine Marie[2]
(1) Inria, Paris, France
(2) Orange Innovation, Caen, France
(3) Sorbonne Université, Paris, France
{lionel.tadonfouet,Eric.De_La_Clergerie}@inria.fr
{fabrice.bourge, tiphaine.marie}@orange.com

RÉSUMÉ

Les conversations d'emails en entreprise sont parfois difficiles à suivre par les collaborateurs car elles peuvent traiter de plusieurs sujets à la fois et impliquer de nombreux interlocuteurs. Pour faciliter la compréhension des messages clés, il est utile de créer des sous-fils de conversations. Dans notre étude, nous proposons un pipeline en deux étapes pour reconnaître les actes de dialogue dans les segments de texte d'une conversation et les relier pour améliorer l'accessibilité de l'information. Ce pipeline construit ainsi des paires de segments de texte transverses sur les emails d'une conversation facilitant ainsi la compréhension des messages clés inhérents à celle-ci. A notre connaissance, c'est la première fois que cette problématique de constitution de fils de conversations est abordée sur les conversations d'emails. Nous avons annoté le corpus d'emails BC3 en actes de dialogues et mis en relation les segments de texte de conversation d'emails de BC3.

ABSTRACT

Email conversations in the workplace are sometimes difficult to follow by collaborators because they can deal with multiple topics and involve many interlocutors. To improve understanding of key messages, it's helpful to create subthreads within the conversation. In our study, we propose a two-stage pipeline to recognize dialogue acts in email text segments and link them to improve information accessibility. This pipeline creates pairs of text segments across the conversation, making it easier to understand the key messages. To our knowledge, this is the first time this issue of creating conversation threads has been addressed in email conversations. We annotated the BC3 corpus of emails with dialogue acts and linked conversation email text segments.

MOTS-CLÉS : fils de conversations, emails, acte de dialogues, appariement d'énoncés, corpus, SetFit.

KEYWORDS: Conversation threads, emails, dialogue acts, utterances pairing, corpus, SetFit.

1 Introduction

Depuis quelques décennies, avec l'évolution d'internet et l'avènement de l'intelligence artificielle, les conversations Médiées par Ordinateur (CMO, *Computer Mediated Communication* – CMC en anglais) sont d'intérêt pour des recherches en linguistique, psychologie et dans bien d'autres domaines. Les

emails, les chats et les échanges dans des forums font partie de ces conversations et sont des canaux d'échanges utilisés dans des entreprises via des outils de communications et de collaboration. Les contenus issus de ces outils regorgent d'importantes connaissances et le fait qu'ils soient peu ou pas structurés limite leur exploitation pour en extraire leur quintessence. Une problématique générale induite par le besoin d'une meilleure compréhension ou l'extraction de connaissances de ces contenus dans le cadre des emails est la reconstruction de fils de conversations d'emails.

Un fil de conversation dans un corpus d'e-mails est formellement défini comme un ensemble d'emails échangés sur le même sujet entre le même groupe de personnes via des actions de réponse ou de transfert (Erera & Carmel, 2008). Pour (Dehghani *et al.*, 2012) il existe deux types de structure de conversation d'emails :

— Linéaire : les emails appartenant à la même conversation sont détectés et disposés dans l'ordre chronologique, formant une structure à une seule branche.
— Arborescente : Dans une conversation, les utilisateurs peuvent choisir de répondre à un email précis déjà existant dans la conversation produisant ainsi une structure en arbre avec une racine et ses branches.

Reconstruire un fil de conversation d'emails consiste ainsi à produire soit la structure linéaire ou arborescente permettant ainsi une meilleure compréhension du contenu de ladite conversation. Plusieurs travaux ont approché la problématique de reconstruction de fils de conversation d'emails sous des trois prismes différents. Tout d'abord, l'algorithme de Zawinski [1] aborde le problème en s'appuyant uniquement sur les méta-données pour la construction de fils de conversation. Ensuite il y a des approches qui se basent sur les contenus afin de regrouper les emails en conversations avec des structures linéaires ou arborescentes. Enfin l'identification des thématiques dans les conversations d'emails sert aussi de base pour une reconstruction de fils de conversations d'emails.

Ces travaux reconstruisent les structures de fils de conversation d'emails permettant une meilleure lisibilité des contenus desdites conversations et une identification des relations parent/enfant entre les emails d'une même conversation. Cependant ils ne permettent pas d'avoir un accès à l'essence des informations contenues dans une conversation. Aussi ces approches ne permettent pas de facilement suivre l'évolution d'une conversation, ni de savoir quelles sont les principales actions menées par les interlocuteurs dans de telles conversations d'emails. Ces actions fortement lieés aux actes de dialogues exprimés dans les messages des interlocuteurs, permettraient de cartographier la progression d'un projet avec en plus les différentes contributions des collaborateurs. Une conversation en plus de permettre des échanges sur des thématiques, est avant tout une communication entre des interlocuteurs, d'où l'existence des actes de dialogue. L'évolution d'une conversation peut par exemple répondre aux questions suivantes : est-ce que les questions posées en amont dans la conversation ont été répondues ou non ; est-ce que des approbations ou désaccords ont été émis en retour à des suggestions exprimées.

Les valeurs ajoutées qui résulteront de la remédiation des insuffisances susmentionnées, constituent les éléments de motivation des travaux décrits dans ce papier. Nous y proposons une approche de constitution de sous-fils de conversation d'emails qui s'appuie sur les métadonnées, principalement la relation **reply-to** entre deux emails, les actes de dialogue de segments de texte extraits d'emails, la similarité sémantique entre ces segments et la production de paires transverses de ces segments de texte. La figure 1 met en avant une conversation du corpus BC3 avec ses emails, ainsi que les paires transverses de segments de texte construites via annotation. Un exemple de relation transverse est la relation entre le segment de texte *"Those who so wish could attend both weeks, and other people could attend only one week"* de l'email 1 qui est une **suggestion** et le **désaccord** *"Jacob, No*

1. message threading

way. Taking one week out of our calender 3 times...". Notre approche de constitution de sous-fils de discussion permet non seulement de démêler de façon fine une conversation d'emails mais aussi surtout de connaître l'état d'évolution de ladite conversation.

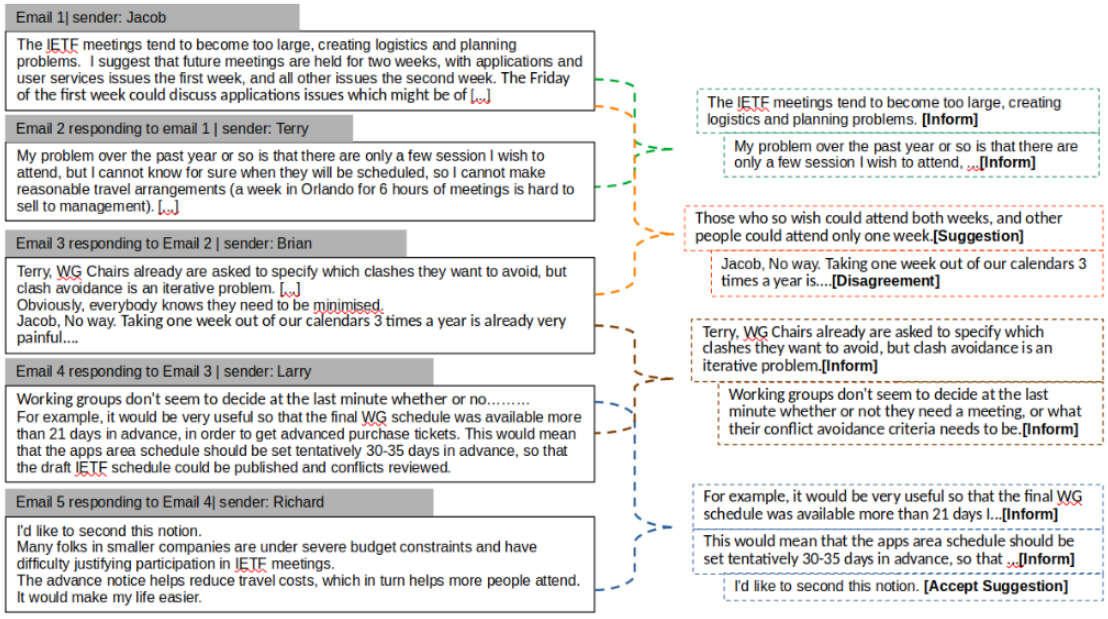


FIGURE 1 – Extrait d'une conversation du corpus BC3 avec des paires de segments de texte appariées

La figure 1 illustre la problématique de constitution de sous-fils de conversations d'emails. Les principales contributions dans ce papier pour la constitution des sous-fils de conversations d'emails sont :

— la production d'un référentiel d'actes de dialogue qui s'appuie sur la norme ISO 24617-2 [2]
— l'utilisation de ce référentiel pour annoter en actes de dialogues les segments de texte du corpus BC3
— La mise en œuvre de deux modèles dont le premier pour la tâche de reconnaissance d'actes de dialogue sur des segments de texte d'emails la seconde pour l'appariement de segments de texte de façon transverse. Le second modèle s'appuie sur les prédictions du premier

2 Travaux connexes

Les emails sont un outil de communication et de collaboration largement utilisé en entreprise et ils contiennent en plus des informations sur l'état d'avancement dans le cadre des projets, des données très riches et difficilement accessibles parce que entremêlées dans des fils de conversations d'emails qui sont peu ou pas structurés. La reconstruction de fils de conversations d'emails et l'identification des thématiques abordés dans ces conversations sont les principales problématiques liées à cette modalité de communication. Pour répondre à ces problématiques, il existe plusieurs approches.

— **Approches basées sur les méta-données pour la construction de fils de conversation**

2. Language resource management -Semantic annotation framework (SemAF)-Part 2: Dialogue acts

L'algorithme de Zawinski est l'un des algorithmes les plus populaires pour la construction de fils de conversation d'emails. Il est basé sur des informations provenant de métadonnées. Cependant, étant donné que ces champs d'en-tête sont facultatifs pour les clients de messagerie, ils ne sont pas toujours disponibles. De plus, ces données ne permettent pas de reconstruire toutes les conversations d'emails avec précision.

— **Approches s'appuyant sur les contenus d'emails afin de regrouper les emails en conversations sans tenir compte de la structure de celles-ci**
(Wu & Oard, 2005) utilisent les objets d'emails pour détecter les fils de conversation. Plus précisément, ils regroupent les emails dans le même fil de conversation si les emails ont le même objet et ont au moins un participant commun. (Wang *et al.*, 2008) extraient les fils de conversation d'emails à l'aide de l'algorithme de Zawinski, puis fusionnent ou décomposent les fils extraits en fonction de leurs objets afin de reconstruire les conversations. De même (Erera & Carmel, 2008) regroupent les emails en conversations cohérentes exploitant une fonction de similarité qui prend en compte tous les attributs d'email pertinents, tels que l'objet de l'email, les participants, la date de soumission et le contenu de l'email.

— **Reconstruction des fils de conversations en structures arborescente**
(Lewis & Knowles, 1997) traitent la reconstruction de fils de conversations d'emails comme un problème d'extraction d'informations. Ils ont étudié cinq approches d'extraction pour déterminer si un email est une réponse à un autre, et leurs résultats indiquent que l'utilisation du texte cité dans un email comme requête et du contenu non cité d'autres e-mails comme documents est la stratégie la plus efficace. Dans leurs recherches, Lewis et Knowles se sont concentrés uniquement sur le corps du texte de l'email et n'ont utilisé aucune autre information disponible dans les emails.
Pour constituer un corpus d'emails pseudo-anonymisés sous des structures arborescentes, (Tadonfouet Tadjou *et al.*, 2021) utilisent les méta-données, les contenus d'emails ainsi que les messages cités dans ceux-ci.
L'approche de (Yeh, 2006) suppose que tous les emails d'une conversation ont le même objet et que la durée de la conversation est généralement plus courte qu'une période fixe. Par conséquent, ils divisent les e-mails en plusieurs groupes, où tous les messages du même groupe ont des objets identiques et la différence de temps maximale entre deux e-mails du groupe est inférieure à un seuil fixe. Ils tentent ensuite de reconstruire l'arborescence des fils de discussion des emails en identifiant les relations parent-enfant entre les emails au sein du même groupe. Bien que leur méthode soit efficace pour détecter les structures arborescentes, les hypothèses qu'ils ont formulées ne sont pas toujours valables, comme en témoignent leurs expériences.
(Joshi *et al.*, 2011) ont utilisé la segmentation et la détection d'emails quasi identiques pour trouver et organiser des messages qui devraient être regroupés en fonction de leurs relations de réponse et de transfert. Ils supposent qu'un email répondant à un autre email contient en tant que texte cité dans un segment séparé, le texte de l'email auquel il répond. Ainsi, ils reconstruisent les fils de conversation tout en tenant compte de ces modèles de segmentation.

(Dehghani *et al.*, 2012, 2013) en s'appuyant sur le corpus BC3 proposent dans le premier papier une approche qui considère la recherche de fils de conversation d'email comme un problème d'optimisation, et exploite la programmation génétique pour rechercher intelligemment dans l'espace des solutions possibles. Dans le second, ils explorent deux nouvelles approches d'apprentissage, LExLinC et LExTreC, qui essayent d'extraire les structures linéaires et arborescentes des conversations, respectivement. LExLinC apprend à extraire les relations

entre fils de conversations d'emails et partitionne l'ensemble des données en clusters d'emails de sorte que chaque cluster représente un thread de conversation. D'autre part, LExTreC essaie d'apprendre des relations parents-enfants parmi les emails et extrait la structure arborescente des conversations.

— **Identification de thématiques dans les conversations d'emails**
(Joty *et al.*, 2010) ont annoté et rendu disponible le corpus BC3 annoté manuellement avec des sujets. Ils évaluent la fiabilité des annotateurs, montrent comment les modèles de segmentation de sujets existants (LCSeg et LDA) peuvent être appliqués aux emails et proposent deux nouvelles extensions de ces modèles qui utilisent non seulement des informations lexicales mais exploitent également une structure de conversation à un niveau plus fin de manière cohérente. Ils capturent la structure de conversation des emails au niveau du fragment (citation) sous la forme d'un graphe de fragment de citations (FQG – *Fragment Quotation Graph*). Un FQG capture la relation de réponse, l'utilisation des citations et d'autres fonctionnalités de conversation. LCSeg proposé pour la première fois par (Galley *et al.*, 2003) est un modèle de segmentation basé sur des chaînes lexicales qui suppose que les changements de sujet sont susceptibles de se produire là où des répétitions fortes de termes commencent et se terminent. Il commence par calculer des chaînes lexicales pour chaque mot qui ne fait pas parti des stop-words basé sur les répétitions de mots. Il classe ensuite les chaînes selon deux mesures : le nombre de mots dans la chaîne et la compacité de la chaîne. Il calcule ensuite la similarité cosinus (ou fonction de cohésion lexicale) à la transition entre les deux fenêtres d'analyse. Une faible similarité indique une faible cohésion lexicale et un changement net signale une forte probabilité d'une frontière de sujet réelle.

3 Méthodologie et Formalisation

3.1 Hypothèse et méthode

Une conversation est constituée d'au moins deux emails, avec au moins deux interlocuteurs, chacun de ses emails aborde au moins un sujet et contient au moins un segment de texte qui est une phrase courte ou une combinaison de plusieurs phrases. Pour mettre en relation certaines phrases d'une conversation d'un email B avec ceux d'une email A, on peut tout simplement s'appuyer sur les métadonnées de la conversation comme la relation *reply-to* entre deux emails, mais aussi sur leur similarité sémantique. Cependant certaines phrases d'emails sont souvent très courtes (moins de 4 mots) et ainsi dépourvues de contexte pour un meilleur score de similarité sémantique avec les phrases d'un email précédent. Une courte phrase pourrait par exemple être *"Ça me va"* : un **accord** ou une *appréciation* qui répond à une *suggestion* dans un précédent email. En général, dans une conversation, certains contenus ou segments de texte à dans un email de continuation sont des réponses, élaborations, suggestions ou d'autres types d'actes de dialogue qui sont en relation avec des segments de texte d'emails précédents. Ces relations entre deux segments de de texte d'emails sont dites **transverses**. On peut aussi dire que deux segments de texte sont adjacents du fait de l'existence d'une relation entre eux. Notre objectif dans ce papier est d'identifier des paires de segments de texte qui sont reliés de manière transverse au sein d'une même conversation. Notre hypothèse est de s'appuyer non seulement sur les métadonnées et la similarité sémantique entre segments de texte mais de les compléter avec les actes de dialogues de ces segments afin d'avoir un système robuste d'appariement de segments de texte entre emails.

Après consolidation des différentes paires entre elles, on obtient des groupes de segments de texte qui

représentent la ou les parties essentielles de la conversation, ce qui vise à en faciliter la lecture et la compréhension. L'extraction de paires transverses d'une conversation se déroule en trois étapes :

— La segmentation des contenus d'emails, par défaut en phrases dans ce papier
— L'assignation d'actes de dialogue aux segments (**Dialogue Acts Recognition - DAR**) : différents modèles sont entraînés basés sur des réseaux de neurones avec sur la dernière couche une fonction SoftMax pour la classification des différents actes de dialogue décrits dans la section 5.1. Le segment de texte précédent celui à prédire est utilisé en contexte afin d'améliorer les performances des modèles.
— L'appariement des segments de texte ou d'énoncés(**AE**) ou des emails d'une conversation : un classifieur binaire est entraîné pour cette tâche en s'appuyant sur le framework SetFit (**?**). Comme entrée à ces modèles, des paires de segments positives et négatives sont construits à partir des corpus existants BC3 et Reddit. Les paires positives et négatives appartiennent à une même conversation. Chaque paire est constituée de deux segments de texte extraits respectivement de deux emails E_i et E_j, i et j des entiers tels que $i < j$. Les deux segments (S_a, S_b) d'une paire positive ont chacun leurs actes de dialogue respectifs (DaS_a, DaS_b) et sont en relation du fait que DaS_b est répond au sens large du terme à DaS_a. Par exemple, DaS_a peut être une suggestion et DaS_b une appréciation. Une paire négative quant à elle a ses segments de texte qui ne sont pas en relation et ont été choisis de façon aléatoire dans une conversation.

3.2 Formalisation du problème

Étant donné une conversation C comportant n emails $\{E_1, E_2, ..., E_n\}$, chacun de ces emails E_i contient m segments de texte $\{s_1^i, s_2^i, ..., s_m^i\}$. La construction des paires de segments de texte

$P = \{(s_a^i, s_b^j), (s_d^k, s_c^l), \cdots (s_m^{n-i}, s_m^n)\}$ transverses sur les emails de la conversation C telle que

$(i < j < k < l \cdots < n)$ et $(a, b, c, d \cdots, m) \in \mathbb{N}$. Les deux éléments d'une paire :

— appartiennent à deux emails distincts E_i et E_j de C avec $i < j$
— ont une relation basée sur les actes de dialogue de type question-réponse, suggestion appréciation, etc.
— peuvent avoir une similarité sémantique

La figure 2 illustre les deux étapes de notre pipeline dont l'objectif est de faire ressortir les paires de segments positives (annotées "yes" dans le schéma)

4 Corpus et annotations

Nous avons utilisé le corpus d'emails BC3 (Ulrich *et al.*, 2008) et le corpus "*Coarse Discourse Sequence Corpus*" (CDSC) de (Zhang *et al.*, 2017) extrait du forum Reddit. CDSC a été annoté en acte de dialogues par trois annotateurs, qui ont aussi mis en relation les différents posts (post B répond au post A) d'une conversation, ce qui a guidé notre choix pour compléter les données du corpus BC3 qui est constitué de seulement 40 conversations pour 261 emails et 1127 phrases. Le corpus BC3 a été construit à la base pour une tâche de résumé de conversations d'emails mais (Jeong *et al.*, 2009) l'ont utilisé dans leurs travaux de classification en actes de dialogues des phrases d'emails et de forums avec des approches semi-supervisées. Ils ont fait réannoter les phrases de BC3 avec douze

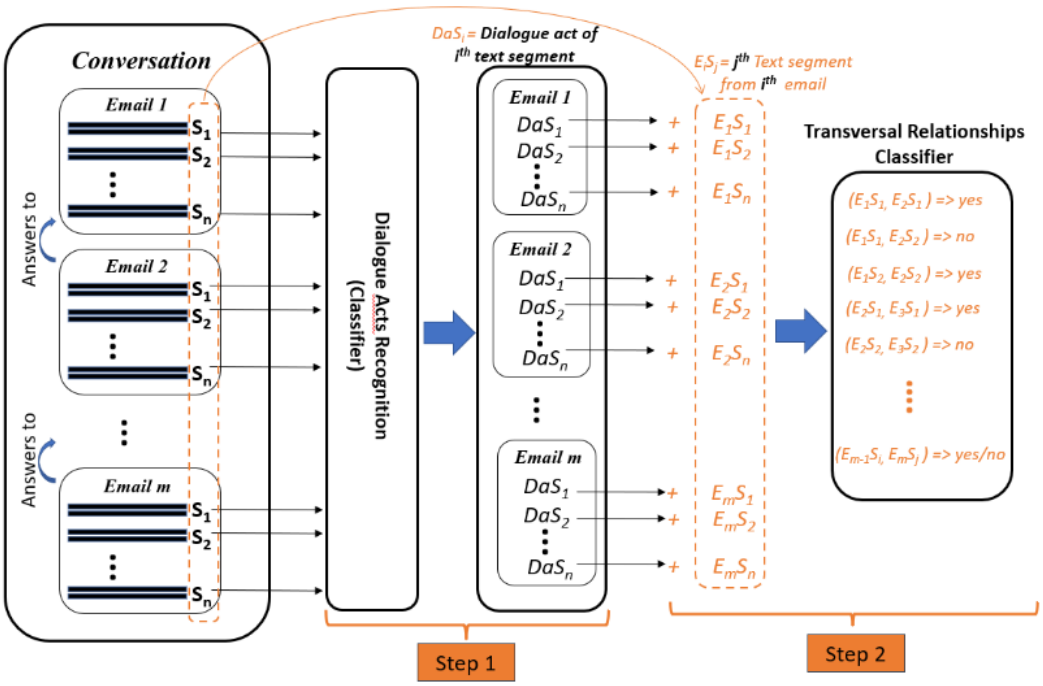


FIGURE 2 – Processus en deux étapes pour l'appariement de segments de texte

actes de dialogues par deux annotateurs avec un accord inter-annotateur égal à 0,79. Cependant nous avons constaté que ces actes de dialogue ne répondaient pas à notre besoin car imprécis : plusieurs phrases d'emails annotées comme *"statement"* étaient pourtant principalement des suggestions ou des élaborations. Pour cette raison, nous avons décidé de réannoter le corpus BC3 en actes de dialogues en relation entre phrases d'emails dans une conversation. Les actes de dialogues utilisés proviennent d'un référentiel que nous avons spécialement établi pour notre besoin sur les conversations d'emails et aussi pour des raisons de clarté.

4.1 Mise en œuvre d'un référentiel d'annotations

La classification de segments de texte en actes de dialogue est une problématique abordée avec d'autres corpus, par exemple avec MRDA [3] un corpus de transcriptions de réunion (Joty & Mohiuddin, 2018; Mohiuddin *et al.*, 2019). Dans la section 5.1 nous décrivons comment nous avons utilisé MRDA pour entraîner des modèles de classification d'énoncés de réunion en actes de dialogue. Afin de pouvoir transférer les modèles de reconnaissance d'actes de dialogues entraînés sur un corpus d'emails, nous avons opté pour une adaptation des actes de dialogue définis dans MRDA parce que ceux-ci sont très diversifiés et certains d'entre-eux peuvent difficilement se retrouver dans les contenus d'emails. L'adaptation que nous avons effectuée s'est déroulé en trois étapes :

— premièrement nous avons identifié sur les différentes fonctions communicatives, les actes de gestion de tâches et les feedbacks définis dans la norme ISO 24617-2 qui répondaient le mieux à l'identification des actes de dialogues dans les emails et aussi aideraient à une structuration dialogique de conversation d'emails

3. Meeting Recorder Dialogue Act Corpus

— ensuite nous avons défini des acronymes que nous avons définis qui permettent de facilement identifier chaque acte de dialogue sans ambiguïté
— enfin nous avons fait correspondre les acronymes d'actes de dialogue du MRDA avec les nôtres en consultant de façon collégiale (deux personnes) une centaine d'énoncés de MRDA de différents types d'actes pour s'assurer de bien tous les prendre en compte.

La table 1 récapitule le référentiel que nous avons défini. Dans la colonne la plus à gauche, les actes de dialogue de MRDA mappés. En orange les fonctions communicatives de la norme ISO 24617-2 et enfin en bleu les acronymes et actes de dialogues que nous avons définis. Les acronymes s'interprètent facilement avec leurs actes de dialogue respectifs, ce qui n'est pas le cas pour ceux de MRDA et de Switchboard Dialogue [4].

		Communicative functions from ISO-24617-2				
MRDA Dialogue Acts (Full Labels)	Ours Labels	0	1	2	3	4
br	Q	Info-seeking functions	Question			
|Q|fh|fh; |Q|q?|e; qy; qo	PrQ			Propositional Question		
bu, d, g	CkQ				Check Question	
qw	SQ			Set Question		
	TQ				Test Question	
qr; qrr|qrr	ChQ			Choice Question		
S; |S|fh|fh	I	Info-providing functions	Inform			
no	An			Answer		
na	Cf				Confirm	
nd; ng	Dcf				Disconfirm	
aa	Ag			Agreement		
ar	Dag			Disagreement		
bc	Cr				Correction	
	AdR	Commissives	Offer	Address Request		
	AcR				Accept Request	
	DeR				Decline Request	
cs	O					
	Pr			Promise		
	AdS			Address Suggestion		
	AcS				Accept Suggestion	
	DeS				Decline Suggestion	
	S	Directives	Suggestion			
co; qy|cs	R		Request			
	Is			Instruct		
	AdO				Address Offer	
	AcO					Accept Offer
	DeO					Decline Offer
fw;by;ft;fa	P		Politeness			
ba	AA		Appreciation/Assessment			
bd	M		Miscellaneous			
df, s|f;bs; arp;bsc; aap; |t	Ex		Elaboration			
am	Hy		Hypothesis, Asumption			

TABLE 1 – Liste d'acronymes définis s'appuyant sur la norme ISO 24617-2 et leur correspondance d'actes de dialogues dans MRDA

4.2 Annotation du corpus BC3

Il existe quelques travaux d'identification d'actes de dialogues dans les conversations d'emails comme ceux de (Taniguchi *et al.*, 2020) qui ont annoté en actes de dialogue plus de 2k fils de conversations du corpus Enron avec deux granularités différentes : 35k phrases avec une granularité fine et 6k emails annotés avec une granularité moins fine. Cependant ce corpus d'emails annotés n'est pas disponible et de par nos travaux, excepté les travaux de (Jeong *et al.*, 2009) pour la tâche de classification des phrases d'emails en actes de dialogues ; il n'existe pas de corpus d'emails finement annoté en actes de dialogue et en relation de messages, ce que nous avons effectué avec le corpus BC3.

Deux personnes ont ainsi annoté ce corpus BC3 en s'appuyant sur le référentiel mis en œuvre. Nous avons annoté 20 conversations, soit 662 segments de texte. La valeur de Kappa (Viera & Garrett, 2005) est de 0.47 pour les annotations en actes de dialogue, cette valeur s'interprète comme un accord modéré entre les deux annotateurs.

Concernant les appariements de segments de texte sur ces 20 conversations, les deux annotateurs ont

4. Switchboard Dialogue Act Corpus

respectivement identifié 289 et 237 relations entre les segments dans lesdites conversations avec une intersection de 107 relations. Ces disparités mettent en exergue la difficulté de la tâche d'appariement de segments de texte transverses même pour des humains. Cependant pour entraîner nos modèles d'appariement de segments de texte, nous avons utilisé l'union des paires positives annotées issues des deux annotateurs, soit 418 au lieu de l'intersection qui est de très petite taille. Nous avons constitué 196 paires négatives avec chaque paire constituée de segments de texte de la même conversation d'emails. Dans la suite de ce papier, BC3 est utilisé pour référencer le total de 614 paires.

4.3 Corpus Reddit

Nous utilisons une version du corpus Reddit [5] (Zhang *et al.*, 2017) finement annoté par trois personnes en actes de dialogue et en relation REPLY-TO entre les messages de chaque conversation. Ces annotations portent sur environ 10k fils de conversation de Reddit. Ce corpus est l'un des rares corpus de conversations asynchrone (forum) annotés sur ces deux aspects.

La table 2 fournit les tailles des différentes paires constituées ainsi que leur distribution pour l'entraînement et les tests de notre modèle. Les actes de dialogues de BC3 et Reddit sont cependant différents dans nos expériences et nous avons donc établi une correspondance des actes de dialogue de BC3 vers ceux de Reddit, obtenant un corpus que nous avons nommé $BC3_{map}$.

DataSet	Train	Validation	Test	Total
BC3	229 (156 PP + 73 PN)	105 (71 PP + 34 PN)	280 (191 PP + 89 PN)	418 PP + 196 PN
Coarse Reddit	7536 (2998 PP + 4538 PN)	932 (374 PP + 558 PN)	942 (375 PP + 567 PN)	3747 PP + 5663 PN
Total	7648 (3110 PP + 4538 PN)	984 (400 PP + 584 PN)	1080 (444 PP + 636 PN)	

TABLE 2 – Distribution des données utilisés (PP : Paires positives, PN : Paires négatives)

5 Expériences, résultats et analyses

5.1 Reconnaissance d'actes de dialogue

Nous utilisons le corpus MRDA afin d'entraîner un modèle pour la classification ou reconnaissance d'actes de dialogue sur des énoncés de conversations. MRDA est à la base un corpus d'échanges audio, d'où la présence de multiples marqueurs de conversations orales tels que *"umh", "umhumh", "you know", "so", "hummm"*, etc. qui sont quasi absents dans les conversations écrites surtout dans des emails d'entreprise. Plusieurs énoncés sont essentiellement constitués de ces marqueurs. Nous sommes partis de l'hypothèse que ces marqueurs vont créer du bruit dans les modèles que nous allons entraîner et donc nous avons filtré le corpus MRDA en supprimant les énoncés constitués seulement de ces marqueurs. Nous avons aussi supprimé ces marqueurs dans les énoncés.

Après avoir filtré le corpus, nous obtenons respectivement 36722, 7985, 7918 énoncés pour les données d'entraînement, de validation et de test. Nous avons fine-tuné le modèle BERT pour la tâche de classification d'énoncés en actes de dialogue en lui rajoutant une couche BiLSTM à chaque fois suivie d'une d'auto-attention ou pas *(BERT_BiLSTM, BERT_BiLSTM_Att)*. Comme entrées à ces modèles, nous utilisons dans un premier temps les énoncés à classifier pris indépendamment les uns des autres et, dans un second temps, nous considérons un contexte qui est l'énoncé précédant

5. Coarse Discourse

immédiatement celui à classifier donnant ainsi lieu au modèle *BERT_BiLSTM_Ctx* par exemple sans la couche d'attention.

Nous avons entraîné certains de ces modèles avec comme entrées des énoncés regroupés, c'est-à-dire que deux ou plusieurs énoncés qui se suivent dans les transcriptions qui sont du même interlocuteur et qui ont le même acte de dialogue sont regroupés en seul énoncé. Ce regroupement donne lieu à des variantes de nos modèles dont les noms se terminent par *GrpFalse* et *GrpTrue* respectivement pour les modèles avec les entrées non groupées et groupées. Nous utilisons les 20 actes de dialogue suivants *['aa', 'adr', 'ads', 'ag', 'an', 'cf', 'cr', 'dag', 'dcf', 'ex', 'hy', 'i', 'is', 'o', 'p', 'pr', 'q', 'r', 's', 'tc']* extraits de la colonne "Our Labels" de la table 1.

5.2 Appariement de segments de texte ou d'Énoncés (AE)

Pour la tâche d'appariement de segments de texte transverses sur les emails d'une conversation, nous utilisons le framework SetFit (SetFit - Efficient Few-shot Learning with Sentence Transformers) de(Tunstall *et al.*, 2022). Nous avons constitué des paires positives et négatives de segments de texte, extraites des corpus BC3 et Reddit.

Les paires positives font partie de la même conversation et le second membre de la paire est en relation avec le premier comme défini dans la section 3.1. Les paires négatives quant à elles sont aussi extraites d'une même conversation, cependant il n'existe aucune relation entre les segments de texte de ces paires. La similarité sémantique est prise en compte avec SetFit qui implémente aussi en son sein l'approche contrastive qui améliore davantage les scores de similarités entre contenus.

Une entrée à notre modèle est donc une paire de segments de texte $([Da]S_i, [Da]S_j)$ avec "[Da]" qui sont des tokens spéciaux créés à partir des actes de dialogues et placés devant chaque segment de texte de la paire. Comme exemple de tokens spéciaux, nous avons [EPNT], [QSTI], AGMT] respectivement pour *"explanation", "question", et "agreement"*

Nous avons entraîné les différents modèles ci-dessous avec pour chacun le (ou les) corpus sur lequel (lesquels) il a été entraîné :

- **AE** : Appariement d'Énoncés sans les actes de dialogues, entraîné afin de montrer quel est l'impact des actes de dialogues sur l'appariement de segment de texte. Ce modèle s'appuie uniquement sur la similarité sémantique
- **AE+ADG** : Appariement d'Énoncés avec les Actes de Dialogues Gold, entraîné de façon indépendante avec Reddit, Reddit+BC3 et Reddit+BC3_map
- **AE+ADP** : Appariement d'Énoncés avec les Actes de Dialogues Prédits en utilisant notre modèle de classification de segments de texte en actes de dialogues. Ce modèle est entraîné de façon indépendante avec Reddit, Reddit+BC3. Les actes de dialogue utilisés reflètent la réalité dans laquelle nos modèles seront utilisés, car les annotations en acte de dialogues seront faites de façon automatique et non manuellement (GOLD) comme dans le modèle AE+ADG

Nous effectuons les tests de nos modèles de différentes manières. D'une part ceux entraînés uniquement sur Reddit sont testés sur les données de test de Reddit et sur l'ensemble du corpus BC3, ceci afin de voir s'il a y transfert de connaissance des données de forums sur les emails. D'autre part les modèles entraînés sur Reddit+ BC3/BC3_map sont uniquement testés sur des leurs données de test respectives extrait du découpage en données d'entraînement, de validation et de test. Ces tests sont effectués sur les données avec et sans actes de dialogue, ceci afin de pouvoir identifier l'apport réel de l'utilisation des actes de dialogues pour notre tâche d'appariement.

Ces modèles ont tous été entraînés avec les mêmes hyperparamètres *({learning_rate :4.3879e -06, num_epochs 5, batch_size :32, 'model_id' :'sentence-transformers/bertbase-nli-mean-tokens', 'num_iterations' : 80})* obtenus en amont en entraînant le modèle **AE+ADG** et ce avec une approche de recherche d'hyperparamètres optimum implémentée avec Optuna.

5.3 Résultats et analyses

5.3.1 Reconnaissance d'actes de dialogue

La figure 3 met en avant les performances des différents modèles que nous avons utilisés pour la classification des segments de texte du corpus MRDA en actes de dialogue. Il ressort de cette figure que le modèle BERT finetuné avec une couche BiLSTM plus une couche d'auto-attention(BERT_BiLSTM_Att_GrpFalse) sans le regroupement des inputs a une meilleure performance au bout de 2 epochs contrairement aux autres combinaisons.

Les modèles qui prennent comme entrée les énoncés avec chacun son contexte respectif donnent de moins bons résultats. Le contexte rajouté à chaque énoncé à classifier devrait logiquement améliorer les performances du modèle, ce qui n'est pas le cas ici. Ceci peut s'expliquer par le fait qu'il y a eu une perte d'information lors de notre processus de filtrage mais aussi du fait que le corpus MRDA est un corpus de transcription de réunion et donc est plutôt constitué de conversations synchrones. De plus lors d'une réunion plusieurs sujets peuvent être abordés de façon entremêlée et donc les contextes que nous rajoutons aux énoncés à classifier n'ont probablement aucune similarité avec ceux-ci au vu de la performance de BERT_BiLSTM_CtxGrpFalse.

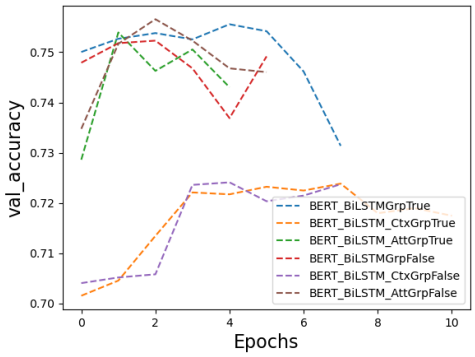


FIGURE 3 – Performances pour la classification en actes de dialogues

5.3.2 Appariement de segments de texte ou d'énoncés (AE)

La table 4 récapitule les scores F1 des différents modèles que nous avons entraînés. Nous avons utilisé la validation croisée k-fold avec k=3 pour les scores F1 lors de l'évaluation de nos modèles sur les données de test BC3, ceci parce que le corpus BC3 est de très petite taille (cf. table 2). Les scores des différents modèles entraînés sur le seul corpus BC3 sont biaisés du fait de la petite de BC3. En plus nous avons relevé de l'overfitting lors de l'entraînement ces modèles.

Les scores des modèles entraînés sur les données avec les actes de dialogues "GOLD" sont meilleurs que ceux entraînés avec les actes prédits. C'est le cas de AE+ADG entraîné sur Reddit+BC3 qui

		AE			AE + ADP			AE + ADG			
	Données d'entrainements	BC3	Reddit	Reddit+BC3	BC3	Reddit	Reddit + BC3	BC3	Reddit	Reddit + BC3	Reddit+BC3_map
F1-score	BC3	0.72	0.61	0.73	0.72	0.60	0.74	0.74	0.62	0.78	0.77
	Reddit	0.52	0.69	0.68	0.50	0.67	0.68	0.50	0.71	0.75	0.73

TABLE 3 – Résultats des modèles d'appariement de segments de texte

donnent respectivement des scores de 0.78 et 0.75 sur les données de test de BC3 et Reddit. Cependant, AE+ADP entraîné aussi Reddit+BC3 donne de moins bons résultats respectivement 0.74 et 0.69 pour ces mêmes données de test, soit une perte d'environ 4 points. Ceci démontre l'importance de l'utilisation des actes de dialogues les plus fiables pour l'appariement de segments de texte dans les conversations. Ainsi plus performant sera notre modèle de classification de segments de texte en actes de dialogue, meilleurs seront les résultats de nos modèles d'appariement.

Dans nos expériences, nous avons mappé les actes de dialogues de BC3 sur ceux de Reddit formant ainsi BC3_map et nous avons entraîné AE+ADG sur Reddit+BC_map qui donne un score de 0.77 sur les données de test de BC3_map contre 0.78 pour le même modèle mais entraîné sur BC3 avec les actes de dialogues "GOLD", cette différence d'un point peut s'expliquer par le fait de la diversité des actes de dialogues que nous avons mis en œuvre dans notre référentiel (section 4.1 soit 20 actes de dialogues) qui améliore les performances de notre modèle.

AE+ADP entraîné respectivement sur Reddit et Reddit+BC3 donne 0.67 et 0.68 lorsqu'évalué sur Reddit. Cette différence d'un point peut s'interpréter par l'ajout des données BC3 (environ 3% de la taille de Reddit) sur le Corpus Reddit. Et on peut en déduire qu'on pourrait gagner des points sur nos scores avec davantage de données.

Ce même modèle AE+ADP entraîné respectivement sur BC3 et Reddit+BC3 a des scores de 0.72 et 0.74 lorsqu'évalué sur BC3, soit une différence de 2 points. Ce gain est dû à l'augmentation de données (Reddit sur BC3), mais traduit aussi le transfert de connaissance des données de forum vers les données d'emails.

Nous avons analysé un échantillon (28%) extrait des données de test de BC3, la majorité des énoncés de cet échantillon sont prédits comme des questions ou des annonces (*inform*). Ci-dessous quelques vrais négatifs (VN) et faux positifs(FP) prédits par le modèle AE+ADP.

1. VN : *Inform : It is not done but you will get the idea. <–> Assessment- it's a good piece of work.*
2. VN : *Inform : If all web authors felt like this about groups they are not prepared to cater to, its no wonder we need WAI. <–> Question : Jonathan, do you really mean to be insulting to me ?*
3. VN : *Inform : Please take a look at [URL] for a first small attempt at this. <–> Inform : Got a could not connect to remote server from both links at [URL]*
4. FP : *Question : My quesiton is how would a screen reader handle that code.... <–> Inform : He just hadn't run into them in the standard version before trying the version for screen reader users.*
5. FP : *Politness : Thanks for the suggestion. <–> Inform : I would skip IE[PATH] since designers worth 2c can tell you already how things work there by reading the code.*
6. FP : *Question : Can you suggest another venue and possible sponsor ? <–> Inform : I want to go to Venice !*

D'une part ces paires d'énoncés font ressortir que nos modèles ont parfois besoin de contexte pour une

meilleure prédiction : un tel contexte pourrait améliorer la classification des paires 1, 2, 4 et 6. D'autre part l'inexactitude des actes de dialogues de certains énoncés contribue à la mauvaise classification des paires qu'elles constituent. Dans la paire 3, les deux énoncés sont en réalité respectivement une requête et une réponse à celle-ci. Le dernier exemple est un classique du type question/réponse. En plus de la petite taille de nos corpus, cette analyse montre les insuffisances de notre approche et identifie clairement les leviers sur lesquelles s'attaquer pour améliorer nos modèles.

Tous les scores de nos modèles avec actes de dialogue "GOLD" (AE+ADG) sont meilleurs que ceux utilisant les actes de dialogue prédits. Cependant l'utilisation concrète de nos modèles en entreprise se fera avec des actes de dialogue prédits et non "GOLD", vu le coût des processus d'annotations.

Comme baseline, nous avons utilisé les métadonnées (reply-to entre les emails d'une conversation) avec d'une part BM25 (Robertson & Zaragoza, 2009), un algorithme de ranking, souvent utilisé comme baseline ou couplé à d'autres méthodes pour la sélection de réponses à un énoncé dans les dialogues (Yan *et al.*, 2018; Chen *et al.*, 2021; Lin *et al.*, 2020; Henderson *et al.*, 2019) et d'autre part avec un système de similarité sémantique basé sur des modèles neuronaux. Pour ce second système, nous utilisons SentenceTransformers (Reimers & Gurevych, 2019).

	Baseline (avec métadonnées)		Modèle entraîné
	BM25	Sentence-BERT :	AE + ADP (entraîné avec Reddit+BC3)
données de test de BC3	0.58	0.65	**0.74**

TABLE 4 – Comparaison de notre modèle avec nos baselines

6 Conclusion

Dans ce papier, nous avons proposé un pipeline qui s'appuie sur la reconnaissance en actes de dialogue de segments de texte et la mise en relation de ceux-ci pour la reconstruction de fils de discussions dans les conversations d'emails. Les résultats de nos modèles d'appariement de segments de texte de conversation d'emails montrent l'intérêt de l'utilisation des actes de dialogues pour ce problème. L'analyse de ces résultats nous a permis d'identifier les insuffisances de notre approche comme l'absence de contexte dans nos énoncées et l'inexactitude des actes de dialogues prédits dans la première étape de notre pipeline. Pour les futurs travaux nous allons combiner les composantes de notre pipeline dans une architecture bout-en-bout qui prendra en entrée une conversation d'emails et produira en sortie de paires de segments de ladite conversation. Cette architecture enrichira les énoncés avec les contextes des leurs conversations respectives.

Suite à une expérimentation préliminaire favorable effectuée lors de nos travaux qui a consisté à segmenter du texte, labelliser les segments obtenus en actes de dialogues et les appariements tout ceci via le prompting avec les larges modèles de langages existants tels GPT-3 (Brown *et al.*, 2020), BLOOM (Workshop *et al.*, 2022), LLAMA (Touvron *et al.*, 2023), nous allons dans nos prochains travaux constituer à partir d'Enron (Klimt & Yang, 2004), un corpus de taille conséquente finement annoté afin d'améliorer les résultats de nos modèles.

Références

BROWN T. B., MANN B., RYDER N. *et al.* (2020). Language Models are Few-Shot Learners. DOI : 10.48550/ARXIV.2005.14165.

CHEN W., GONG Y., XU C., HU H., YAO B., WEI Z., FAN Z., HU X., ZHOU B., CHENG B. *et al.* (2021). Contextual fine-to-coarse distillation for coarse-grained response selection in open-domain conversations. *arXiv preprint arXiv :2109.13087*.

DEHGHANI M., ASADPOUR M. & SHAKERY A. (2012). An Evolutionary-Based Method for Reconstructing Conversation Threads in Email Corpora. ASONAM '12, p. 1132–1137, USA : IEEE Computer Society. DOI : 10.1109/ASONAM.2012.195.

DEHGHANI M., SHAKERY A., ASADPOUR M. & KOUSHKESTANI A. (2013). A learning approach for email conversation thread reconstruction. *Journal of Information Science*, **39**, 846 – 863.

ERERA S. & CARMEL D. (2008). Conversation Detection in Email Systems. In C. MACDONALD, I. OUNIS, V. PLACHOURAS, I. RUTHVEN & R. W. WHITE, Éds., *Advances in Information Retrieval*, p. 498–505, Berlin, Heidelberg : Springer Berlin Heidelberg.

GALLEY M., MCKEOWN K. R., FOSLER-LUSSIER E. & JING H. (2003). Discourse Segmentation of Multi-Party Conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, p. 562–569, Sapporo, Japan : Association for Computational Linguistics. DOI : 10.3115/1075096.1075167.

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. *arXiv preprint arXiv :1906.01543*.

JEONG M., LIN C.-Y. & LEE G. G. (2009). Semi-supervised Speech Act Recognition in Emails and Forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, p. 1250–1259, Singapore : Association for Computational Linguistics.

JOSHI S., CONTRACTOR D., NG K., DESHPANDE P. & HAMPP T. (2011). Auto-grouping emails for faster e-discovery. *Proceedings of the VLDB Endowment*, **4**, 1284 – 1294.

JOTY S., CARENINI G., MURRAY G. & NG R. T. (2010). Exploiting Conversation Structure in Unsupervised Topic Segmentation for Emails. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, p. 388–398, Cambridge, MA : Association for Computational Linguistics.

JOTY S. & MOHIUDDIN T. (2018). Modeling Speech Acts in Asynchronous Conversations : A Neural-CRF Approach. *Computational Linguistics*, **44**(4), 859–894. DOI : 10.1162/coli_a_00339.

KLIMT B. & YANG Y. (2004). The Enron Corpus : A New Dataset for Email Classification Research. In J.-F. BOULICAUT, F. ESPOSITO, F. GIANNOTTI & D. PEDRESCHI, Éds., *Machine Learning : ECML 2004*, p. 217–226, Berlin, Heidelberg : Springer Berlin Heidelberg.

LEWIS D. D. & KNOWLES K. A. (1997). Threading electronic mail : A preliminary study. *Information Processing Management*, **33**(2), 209–217. Methods and Tools for the Automatic Construction of Hypertext, DOI : https ://doi.org/10.1016/S0306-4573(96)00063-5.

LIN Z., CAI D., WANG Y., LIU X., ZHENG H.-T. & SHI S. (2020). The world is not binary : Learning to rank with grayscale data for dialogue response selection. *arXiv preprint arXiv :2004.02421*.

MOHIUDDIN T., NGUYEN T.-T. & JOTY S. (2019). Adaptation of Hierarchical Structured Models for Speech Act Recognition in Asynchronous Conversation. 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. 1326–1336, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1134.

REIMERS N. & GUREVYCH I. (2019). Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks. *CoRR*, **abs/1908.10084**.

ROBERTSON S. & ZARAGOZA H. (2009). The Probabilistic Relevance Framework : BM25 and Beyond. *Found. Trends Inf. Retr.*, **3**(4), 333–389. DOI : 10.1561/1500000019.

TADONFOUET TADJOU L., BOURGE F., MARIE T., ROMARY L. & DE LA CLERGERIE É. (2021). Building A Corporate Corpus For Threads Constitution. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, p. 193–202, Online : INCOMA Ltd.

TANIGUCHI M., UEDA Y., TANIGUCHI T. & OHKUMA T. (2020). A Large-Scale Corpus of E-mail Conversations with Standard and Two-Level Dialogue Act Annotations. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4969–4980, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.436.

TOUVRON H., LAVRIL T., IZACARD G., MARTINET X., LACHAUX M.-A., LACROIX T., ROZIÈRE B., GOYAL N., HAMBRO E., AZHAR F., RODRIGUEZ A., JOULIN A., GRAVE E. & LAMPLE G. (2023). LLaMA : Open and Efficient Foundation Language Models. DOI : 10.48550/ARXIV.2302.13971.

TUNSTALL L., REIMERS N., JO U. E. S., BATES L., KORAT D., WASSERBLAT M. & PEREG O. (2022). Efficient Few-Shot Learning Without Prompts. DOI : 10.48550/ARXIV.2209.11055.

ULRICH J., MURRAY G. & CARENINI G. (2008). A Publicly Available Annotated Corpus for Supervised Email Summarization.

VIERA A. J. & GARRETT J. M. (2005). Understanding interobserver agreement : the kappa statistic. *Family medicine*, **37**(5), 360—363.

WANG X., XU M., ZHENG N. & CHEN M. (2008). Email Conversations Reconstruction Based on Messages Threading for Multi-person. *2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing*, **1**, 676–680.

WORKSHOP B., :, SCAO T. L., FAN A., AKIKI C. *et al.* (2022). BLOOM : A 176B-Parameter Open-Access Multilingual Language Model. DOI : 10.48550/ARXIV.2211.05100.

WU Y. & OARD D. W. (2005). Indexing emails and email threads for retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

YAN Z., DUAN N., BAO J., CHEN P., ZHOU M. & LI Z. (2018). Response selection from unstructured documents for human-computer conversation systems. *Knowledge-Based Systems*, **142**, 149–159.

YEH J.-Y. (2006). Email Thread Reassembly Using Similarity Matching. In *International Conference on Email and Anti-Spam*.

ZHANG A., CULBERTSON B. & PARITOSH P. (2017). Characterizing Online Discussion Using Coarse Discourse Sequences.

Deuxième partie

Articles courts

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

Intégration du raisonnement numérique dans les modèles de langue : État de l'art et direction de recherche

Sarah Abchiche[1, 2] Lynda Said Lhadj[1] Laure Soulier[2, 3] Vincent Guigue[2, 4]
(1) Ecola Nationale Supérieure d'Informatique (ESI), LCSI, Alger, Algérie
(2) Sorbonne Université, CNRS, ISIR, F-75005 Paris, France.
(3) Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400, Orsay, France.
(4) AgroParisTech, UMR MIA-PS, France.
is_abchiche@esi.dz, l_said_lhadj@esi.dz, laure.soulier@isir.upmc.fr, vincent.guigue@isir.upmc.fr

RÉSUMÉ

Ces dernières années, les modèles de langue ont connu une évolution galopante grâce à l'augmentation de la puissance de calcul qui a rendu possible l'utilisation des réseaux de neurones. Parallèlement, l'intégration du raisonnement numérique dans les modèles de langue a suscité un intérêt grandissant. Bien que l'entraînement des modèles de langue sur des données numériques soit devenu un paradigme courant, les modèles actuels ne parviennent pas à effectuer des calculs de manière satisfaisante. Pour y remédier, une solution est d'entraîner les modèles de langue à utiliser des outils externes tels qu'une calculatrice ou un "runtime" de code python pour effectuer le raisonnement numérique. L'objectif de ce papier est double, dans un premier temps, nous passons en revue les travaux de l'état de l'art sur le raisonnement numérique dans les modèles de langue et dans un second temps nous discutons des différentes perspectives de recherche pour augmenter les compétences numériques des modèles.

ABSTRACT

Integrating numerical reasoning into language models : State of the art.

In recent years, language models have undergone a rapid evolution thanks to the increase in computing power that has made the use of neural networks possible. At the same time, the integration of numerical reasoning in language models has attracted growing interest. Thus, training language models on numerical data has become a common paradigm, although current models have not been able to perform well in terms of calculations. To overcome this limitation, one solution is to train language models to use external tools such as a calculator or a python code runtime to perform numerical reasoning. This paper has two objectives, first we review state of the art work that has incorporated numerical reasoning and second we discuss different research perspectives to increase the numerical performance of language models.

MOTS-CLÉS : Modèles de langue, raisonnement numérique, question réponses.

KEYWORDS: Language models, numerical reasoning, question answering.

1 Introduction

Le traitement automatique du langage naturel (TAL) a été boulversé ces dernières années grâce aux Transformers (Vaswani *et al.*, 2017). Cette architecture permet d'encoder efficacement le sens des

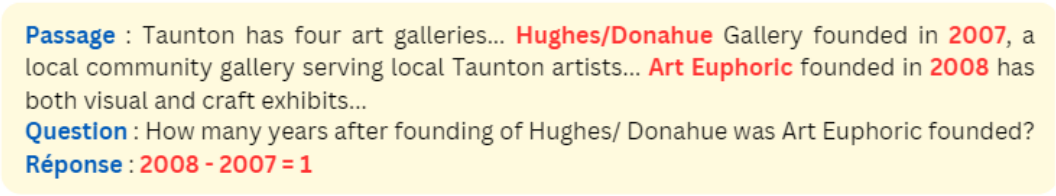
mots et des paragraphes avec leur contexte dans un espace vectoriel de grande dimension à travers un mécanisme d'attention qui focalise le modèle sur certains groupes de mots. Ce mécanisme a démontré des performances importantes dans de nombreuses tâches de TAL comme les tâches de séquence à séquence (e.g., traduction automatique (Luong *et al.*, 2015), résumé de documents (Rush *et al.*, 2015)) et les tâches de classification (e.g., l'analyse de sentiments (Basiri *et al.*, 2021), l'étiquetage des parties du discours ou la catégorisation thématique (Cheng *et al.*, 2019)). Cette avancée a permis aux modèles de langue pré-entraînes (*Pre Trained Language Models*, PTLM) de comprendre les nuances et les subtilités du langage naturel devenant ainsi des outils universels (Howard & Ruder, 2018) utilisés dans de nombreux domaines (Araci, 2019; Steinberg *et al.*, 2021; Chen *et al.*, 2022; Borsos *et al.*, 2022). Un exemple marquant est le système conversationnel ChatGPT qui est conçu pour répondre à des questions très diverses et résoudre toute une série de problèmes en communiquant aux utilisateurs la réponse d'une manière semblable à celle d'un être humain (Jiao *et al.*, 2023; Lund & Wang, 2023). BLOOM, avec 176 milliards de paramètres, est capable de générer du texte dans 46 langues et 13 langages de programmation (Scao *et al.*, 2022) tandis que LaMDA saisit des nuances fines distinguant le dialogue des autres formes de langage (Thoppilan *et al.*, 2022). Aujourd'hui, grâce aux PTLM, souvent il suffit d'étiqueter une petite quantité de données et d'y ajuster un modèle de langue pré-entraîné pour résoudre un problème (Houlsby *et al.*, 2019), en effet, celui-ci ayant déjà acquis une quantité importante des connaissances en TAL au préalable. Le succès des PTLM est également lié à l'explosion des données textuelles disponibles sur internet, qui ont permis de pré-entraîner ces modèles sur des corpus de plus en plus vastes et variés incluant l'intégralité de Wikipédia, Reddit et de nombreuses sources d'informations. Par exemple, le corpus C4 (Raffel *et al.*, 2020) atteint aujourd'hui la taille record de 800 Go de textes et de méta données. Cependant, les compétences de haut niveau, telles que la capacité à effectuer un raisonnement numérique sur du texte, restent un verrou lorsque ces modèles sont appris seulement avec un objectif de masquage des mots ou des générations de phrases contextuelles (Dua *et al.*, 2019; Andor *et al.*, 2019; Chen *et al.*, 2020).

Le raisonnement numérique est une tâche critique qui intervient dans divers scénarios allant du shopping à la modélisation du climat (Lithner, 2000). Il s'agit de traiter des opérations numériques telles que l'addition ou la multiplication et d'interpréter les tendances et les relations numériques (Cobbe *et al.*, 2021). En effet, des données numériques sont présentes dans la grande majorité des documents (finance, santé, journalisme, etc) (Gelman & Butterworth, 2005), d'où le besoin d'avoir des modèles de langue compétents en raisonnement numérique (Dua *et al.*, 2019). Néanmoins, le raisonnement numérique sur du texte est une tâche particulièrement difficile (Al-Negheimish *et al.*, 2021; Mialon *et al.*, 2023) car le modèle doit 1) comprendre les scénarios complexes décrits dans les textes des problèmes, 2) trouver l'enchaînement des opérations nécessaires, 3) identifier les variables mathématiques et associer le texte à la logique des équations mathématiques, 4) projeter les données vers un espace sémantique (ou un système) différent pour effectuer les calculs et enfin 5) produire le résultat attendu et/ou générer la réponse en langage naturel.

La figure 1 montre un exemple de questions à partir de DROP (Dua *et al.*, 2019). Pour répondre à la question "How many years after founding of Hughes/ Donahue was Art Euphoric founded ?", le modèle doit d'abord comprendre le contexte décrit et identifier les années de création des deux entreprises (Hughes/Donahue et Art Euphoric). Ensuite, il doit trouver l'opération à effectuer, dans ce cas c'est une soustraction pour enfin générer la réponse qui est un.

Dans cet article nous nous intéressons à la problématique d'intégration du raisonnement numérique dans les modèles de langue. En d'autres terme, l'enjeu est de résoudre des problèmes mathématiques décrit en langage textuel en utilisant les modèles de langues.

FIGURE 1 – Exemple de question de DROP (Dua *et al.*, 2019) nécessitant un raisonnement numérique



1. Nous proposons un état de l'art des différentes approches d'incorporation des compétences numériques dans les modèles de langues.
2. Nous discutons des perspectives de recherche pour construire des modèles capables d'identifier un enchaînement d'opérations numériques et d'extraire toutes les informations nécessaires pour l'effectuer.

2 État de l'art

Nous présentons les principales approches permettant aux modèles d'effectuer des raisonnements numériques sur du texte. Nous distinguerons d'une part les architectures spécifiques basées sur des modules de raisonnement ou de prédiction et, d'autre part, les larges modèles de langues pré-entraînés pour effectuer des calculs ou pour générer des programmes de raisonnement permettant d'arriver à la réponse. Pour évaluer la capacité de ces modèles à résoudre des questions nécessitant un calcul arithmétique, les approches actuelles reposent sur la métrique "Exact Match". Ainsi, il est question de générer une réponse pour la comparer à la réponse correcte.

2.1 Architectures spécifiques au raisonnement numérique

Les premiers modèles développés pour résoudre des problèmes de raisonnement numérique à partir de texte utilisent des architectures spécialisées dotées de modules de raisonnement.

Le modèle NAQANet, basé sur le modèle QANet (Yu *et al.*, 2018), a été proposé pour intégrer le raisonnement à travers un module de prédiction permettant la génération de quatre types de réponses : l'extraction de texte, les comptages, les additions ou les soustractions de nombres sous forme d'expression arithmétique. De nombreux travaux ont suivi la même approche pour la génération des réponses tels que (Ran *et al.*, 2019; Geva *et al.*, 2020; Chen *et al.*, 2020; Zhou *et al.*, 2022). Parmi ces approches, plusieurs sont basées sur des graphes pour modéliser les dépendances entre valeurs numériques. Par exemple, NumNet de (Ran *et al.*, 2019), construit un graphe à partir des nombres mentionnés dans la question et le passage textuel. Ces derniers représentent les noeuds du graphe et leurs relations encodent les comparaisons entre eux. L'architecture de NumNet atteint de meilleures performances que le modèle NAQANet en développant un cadre qui permet une comparaison numérique des nombres. Néanmoins, de nombreuses questions impliquent la prise en compte d'un nombre intermédiaire n'apparaissant pas littéralement dans le document ou la question, ce qui limite la performance de NumNet. La table 1 montre un exemple de ce type de questions à partir de DROP (Dua *et al.*, 2019), le modèle doit d'abord effectuer des additions pour trouver le pourcentage de personnes ayant un âge supérieur à 40 et le pourcentage de personne ayant un âge inférieur à 19 pour ensuite les comparer afin de retourner la catégorie ayant le plus grand pourcentage. A droite, la réponse générée par NumNet est une preuve de l'inefficacité de ce modèle.

Dans une approche plus élaborée, (Chen *et al.*, 2020) utilisent un graphe orienté qui encode les

TABLE 1 – Exemple de question de DROP (Dua *et al.*, 2019) nécessitant un calcul intermédiaire

Question	Passage	Réponse	NumNet
Were more people 40 and older or 19 and younger?	Of Saratoga Countys population in 2010, 6.3% were between ages of 5 and 9 years, 6.7% between 10 and 14 years, 6.5% between 15 and 19 years, ... , 7.9% between 40 and 44 years, 8.5% between 45 and 49 years, 8.0% between 50 and 54 years, 7.0% between 55 and 59 years, 6.4% between 60 and 64 years, and 13.7% of age 65 years and over ...	40 and older	19 and younger

relations entre les nombres et les entités mentionnés dans le contexte et la question. Dans l'exemple de la table1, les nombres 5 et 9 sont du même type (age) et ils sont liés à l'entité *year*. Le modèle construit un réseau d'attention du graphe qui incorpore l'encodage contextuel de la question dans le processus de raisonnement.

Le développement de structures spécialisées destinées au raisonnement numérique est un premier pas important dans l'amélioration des compétences numériques des modèles. Toutefois, ces modèles présentent des limitations en raison de leur incapacité à effectuer des calculs complexes. Par exemple, certains modèles ont été conçus pour compter jusqu'à neuf ou pour effectuer des opérations telles que l'addition et la soustraction seulement à partir des nombres vus en apprentissage : si le résultat implique plus de valeurs neuves ou que les nombres n'apparaissent pas dans les données d'entraînement le modèles est incapable d'y répondre (Dua *et al.*, 2019; Ran *et al.*, 2019; Chen *et al.*, 2020; Zhou *et al.*, 2022). L'étude expérimentale réalisée par (Al-Negheimish *et al.*, 2021) pour tester les compétences numériques des modèles de langue a montré que les modèles précédents ne parviennent pas nécessairement à la bonne réponse en exploitant les bonnes informations. Ces expérimentations ont mis en lumière des lacunes de ces modèles : ces approches sont souvent capables de répondre aux questions sans avoir accès aux contextes ! Ils captent des motifs récurrents dans le jeu de données pour extrapoler la réponse : ils sont donc biaisés par rapport au format des passages et des questions ou à la distribution des réponses. Ainsi, il existe une grande disparité de performance entre les questions dont les réponses sont les plus fréquentes par rapport aux autres.

2.2 Modèles de langues pré-entraînés augmentés de compétences numériques

Dans cette section, nous nous intéresserons aux travaux explorant le pré-entraînement des modèles de langue afin de générer les résultats numériques souhaités directement, à travers des techniques comme l'augmentation de données textuelles ou numériques (Geva *et al.*, 2020; Yang *et al.*, 2021), le pré-entraînement séquentiel sur plusieurs jeux de données ou l'utilisation de programmes et de leurs résultats comme données d'entraînement (codes, requêtes, etc.) comme dans (Pi *et al.*, 2022).

GENBERT de (Geva *et al.*, 2020) est une amélioration directe du modèle BERT (Devlin *et al.*, 2018). Le module de prédiction est similaire à celui de NaQANet, où chaque type de réponse attendue est traité séparément. Le modèle est d'abord pré-entraîné sur un objectif classique de modélisation de langue puis sur des données numériques –sous forme d'expressions arithmétiques associées à leurs résultats– pour apprendre au modèle à effectuer des calculs. Enfin, le modèle est spécialisé sur des questions-réponses pour apprendre à générer les sorties attendues. Le modèle pré-entraîné se veut générique : il peut être adapté à différents jeux de données tel que DROP (Dua *et al.*, 2019). GENBERT atteint des performances similaires à l'état de l'art sur DROP (à taille de modèle comparable) et il s'adapte aux jeux de données de problèmes mathématiques (MWP) (Koncel-Kedziorski *et al.*, 2016; Hosseini *et al.*, 2014; Roy *et al.*, 2015; Koncel-Kedziorski *et al.*, 2015), tout en maintenant des

performances élevées sur SQuAD (Rajpurkar *et al.*, 2016).

Dans le même contexte, (Yang *et al.*, 2021) comparent cinq pipelines d'entraînement séquentiel qui adaptent un modèle T5 pré-entraîné pour le raisonnement numérique sur du texte. Chaque pipeline comprend deux étapes : pré-entraînement sur des données de raisonnement numérique et de compréhension générale de texte, suivi d'un réglage fin sur DROP (Dua *et al.*, 2019) et d'une tâche de classification dérivée de DROP où il s'agit de classifier le type de la réponse. L'entraînement multi-tâches est adapté à chaque étape en utilisant différents ensembles de données (données numériques et textuelles synthétiques (Geva *et al.*, 2020), DROP et SQuAD).Le pipeline d'apprentissage séquentiel nécessite peu de ressources, il permet de tester différentes hypothèses et donne de bonnes performances même en utilisant des modèles de taille limitée (T5-small). Cette approche a permis de réaliser de grandes améliorations en termes de raisonnement numérique par rapport à un modèle T5 de base.

Dans une autre optique, POET "Program Executor" (Pi *et al.*, 2022) est un nouveau paradigme de pré-entraînement qui permet aux modèles de langue d'acquérir les *capacités de raisonnement* des exécuteurs de programmes en s'entraînant sur des données composées de programmes associées à leurs résultats d'exécution. POET est conceptuellement simple et peut être instancié sur différents types de programmes : POET-Math, POET-Logic et POET-SQL. L'idée est que les modèles de langues, pour prédire le résultat d'un programme, apprennent à imiter les procédures d'exécution de programmes : ils pourraient ensuite potentiellement apprendre plus facilement les mécanismes de raisonnement logique que les humains ont adoptées pour créer l'exécuteur de programme. POET a été testé avec succès à partir de BART et RoBERTa, après une phase d'adaptation, sur DROP, HotpotQA (Yang *et al.*, 2018), TAT-QA (Zhu *et al.*, 2021) et EQUATE (Ravichander *et al.*, 2019).

Dans cette classe de modèles, la majorité des erreurs se produisent sur des exemples qui exigent des compétences de raisonnement qui ne sont pas couvertes par l'ensemble de données de pré-entraînement (Geva *et al.*, 2020; Yang *et al.*, 2021). Bien que POET ait appris des compétences de raisonnement des exécuteurs de programmes, il ne peut répondre qu'aux tâches similaires à celles de l'ensemble de données de pré-entraînement. Pour combler les lacunes de ces modèles, une option serait d'augmenter l'ensemble de données de pré-entraînement pour élargir l'éventail des raisonnements complexes assimilés et mieux répondre aux questions des jeux de données. Malgré une approche assez générique, le problème de la généralisation aux nouvelles données reste un défi.

2.3 Modèles de langues pour la génération de programmes de raisonnement

Dans cette section, nous nous intéressons à une troisième voie émergeante qui consiste à générer un ensemble d'instructions –ie un programme– qui sera interprété par un calculateur externe. La production du résultat final (e.g. manipulation des nombres, calculs) est ainsi déléguée à un outil existant, comme présenté dans la figure 2. (Mialon *et al.*, 2023) ont effectué une étude très large sur l'augmentation des modèles de langues avec des compétences basées des outils externes.

Le modèle présenté dans (Andor *et al.*, 2019) permet au modèle BERT d'effectuer un raisonnement numérique léger avec un pré-entraînement sur DROP (Dua *et al.*, 2019). BERT est augmenté avec un ensemble prédéfini de programmes exécutables qui englobent l'arithmétique simple ainsi que l'extraction de texte. Plutôt que d'avoir à apprendre à manipuler les nombres directement (Geva *et al.*, 2020; Yang *et al.*, 2021; Pi *et al.*, 2022; Xue *et al.*, 2022), le modèle peut générer un programme et l'exécuter. Les programmes sont simples de la forme *Opération(argument, ...)* parmi l'espace des dérivations défini par les auteurs. Cet espace comprend des expressions (yes, no, unkown, 0,...,9), des opérations numériques (diff, mul, div, sum et diff100), des opérations d'extraction de texte (span)

FIGURE 2 – Modèles de langues et modèles de langues augmentés d'outils, (Parisi *et al.*, 2022).

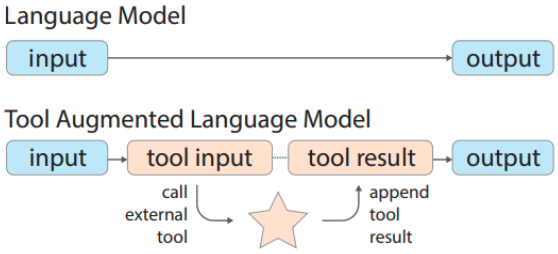


FIGURE 3 – Exemples avec Bhaskara en calcul et algèbre sur le jeu de données LILA (Mishra *et al.*, 2022). Le programme est souvent plus pertinent que l'estimation numérique.

Task: Basic Math
Problem: Before December, customers buy 1346 ear muffs from the mall. During December, they buy 6444, and there are none. In all, how many ear muffs do the customers buy?

Predicted Answer: 1346.0 ✗
Generated Program:
```
answer = 1346.0 + 6444.0
print (answer)
# Result == > 7790.0
```

Gold Answer: 7790.0 ✓

Task: Linear Algebra
Problem: Find the determinant of the matrix

0	-2	-3
0	5	0
1	3	2

Predicted Answer: - 8 ✗
Generated Program:
```
import numpy as np
a = np.array([[0 , -2 , -3], [0 , 5 , 0], [1 , 3 , 2]])
print(np.linalg.det(a))
# Result == > 15.0
```

Gold Answer: 7790.0 ✓

et des compositions (merge, sum3). Une fois l'opération arithmétique générée, le calcul est exporté vers un calculateur externe. Cette approche dépasse l'état de l'art sur certaines tâches spécifiques, néanmoins, elle est très préliminaire dans le domaine et ne couvre pas les calcul numériques avancés.

Dans le même sens, (Mishra *et al.*, 2022) explore les performances de deux types de modèles nommés BHASKARA (P et A). Le premier génère des programmes python qui sont ensuite exécutés afin de prédire le résultat numérique attendu (BHASKARA-P) : le modèle n'est pas entraîné pour exécuter les calculs en interne mais plutôt pour planifier et générer les différentes étapes qui permettront d'atteindre le résultat souhaité. Le second modèle est entraîné pour produire le résultat final directement (BHASKARA-A), tous les calculs se font en interne dans le modèle. La figure 3 montre des exemples de programmes générés et de résultats estimés par BHASKARA. Les résultats de l'expérimentation ont montré que la génération de programme surpasse nettement la prédiction directe des résultats. L'étude de ce travail nous a permis de conclure que les modèles de langue manipulent mieux les opérateurs de haut niveau et leur enchaînement que les calculs numériques directs. De plus, le modèle est capable d'appeler des bibliothèques externes pour effectuer des calculs avancés de manière pertinente. Par exemple, le modèle utilise `scipy.stats.entropy` ou `np.linalg.det` respectivement en statistique et en algèbre linéaire lors de la résolution de problèmes.

Très récemment, (Schick *et al.*, 2023) ont introduit Toolformer, un modèle qui a été entraîné pour apprendre à utiliser différents outils externes tels qu'une calculatrice, un système de questions-réponses, un moteur de recherche, un système de traduction et un calendrier. Le modèle est capable de décider quel outil appeler, quand l'appeler, avec quels arguments lancer la requête et comment incorporer au mieux les résultats dans la prédiction future de texte. Toolformer ne peut pas (encore) gérer les calculs complexes en raison de la nature des exemples utilisés pour l'entraîner à exploiter la calculatrice. L'ensemble de données textuelles ne contient que des documents plutôt simples, qui remplissent l'une des conditions suivantes : (i) contiennent au moins trois nombres dans une fenêtre de 100 mots, où l'un de ces nombres est le résultat d'une opération mathématique appliquée aux deux

autres, (ii) contiennent l'une des séquences "=", "égale", "égale à", "total de", "moyenne de" suivie d'un nombre, ou (iii) contiennent au moins trois nombres. Il s'agit néanmoins d'une proposition intéressante qui ouvre beaucoup de perspectives.
Avec une philosophie très proche, (Lyu *et al.*, 2023) propose Faithful CoT *Chain of Thought*. Il s'agit d'un cadre de prompting décomposant une tâche de raisonnement en deux étapes : 1) la traduction (requête en langage naturel → chaîne de raisonnement symbolique) et 2) la résolution du problème (chaîne de raisonnement → réponse), en utilisant respectivement un PTLM pour générer la chaîne de raisonnement et un outil pour l'exécuter.

3 Discussion et perspectives

Dans cette section, nous discutons des perspectives de recherche que nous envisageons pour proposer des modèles de langues dotés de compétences numériques. L'idée de planifier et générer les différentes étapes du raisonnement nous semble très prometteuse (Andor *et al.*, 2019; Mishra *et al.*, 2022). En effet, le modèle de langue est très performant sur cette tâche tandis que ses faiblesses calculatoires peuvent être palliées en faisant appel à un calculateur externe. L'enjeu est alors d'apprendre aux modèles de langage à utiliser des outils externes (Parisi *et al.*, 2022; Schick *et al.*, 2023). La richesse des outils et bibliothèques existants pour des problèmes très divers ouvre de nombreuses perspectives. Accessoirement le comportement des modèles de langue se rapprocherait alors de celui des humains qui font appel quotidiennement à une large palette d'outils pour résoudre des problèmes divers.

3.1 Capacité des modèles de langues à générer des chaînes de raisonnement

Actuellement, les modèles de langage ont des lacunes sur le raisonnement numérique et sont incapables d'effectuer des calculs complexes. Une idée serait de se concentrer sur la capacité de génération de la chaîne de raisonnement, avec et sans transfert, en déléguant le calcul numérique à des opérateurs externes. Avec transfert en ajustant les modèles de langues de génération de code disponibles sur les datasets de raisonnement comme DROP et sans transfert en utilisant les techniques de prompting. La catégorisation des raisonnements et la supervision de ces chaînes de raisonnement sont réalisables à partir des jeux de données existants et pourraient bénéficier de l'augmentation de données. Une clé réside dans la capacité à créer des fonctions de coût intermédiaires caractérisant les enchaînements locaux d'opérations, à la manière de ce qui existe en apprentissage par renforcement. Le résultat final dépendant directement de calculateur externe serait alors utilisé dans les métriques d'évaluation plus que dans la stratégie d'apprentissage elle-même.

La génération des chaînes de raisonnement ou processus de raisonnement sous formes de codes Python entre parfaitement dans le cadre défini précédemment : 1) un code est essentiellement un enchaînement d'instructions sous forme de texte, la modalité idéale pour les modèles de langue. Il existe d'ailleurs déjà des modèles de langues entraînés sur du code qui pourraient servir de base à ce travail (Black *et al.*, 2022; Nijkamp *et al.*, 2022). 2) Il est possible de résoudre toute sorte de problèmes en utilisant les langages de programmation avec des outils de plus ou moins haut niveau selon les bibliothèques considérées. Ainsi, il s'agit d'un cadre pertinent pour étudier les capacités de généralisation en raisonnement numérique. Contrairement à Toolformer (Schick *et al.*, 2023) qui augmente les PTLMs en utilisant plusieurs outils, nous nous concentrerons sur la partie raisonnement numérique directement dans le langage de programmation : nos outils seront des fonctions.

Au bout de la chaîne, pour pouvoir produire le texte présentant les résultats des raisonnements

numériques générés, nous pensons repartir de la littérature data-to-text : l'enjeu est d'abord de placer le résultat calculé dans une phrase pertinente vis-à-vis de la question et du contexte puis d'exploiter la génération de texte pour expliquer les grandes étapes du raisonnement construit précédemment.

3.2 Application à des tâches de génération de texte à partir de données structurées ("data-to-text generation")

Le raisonnement est présent implicitement dans de nombreuses tâches de TAL (question-réponses, génération de texte, ...). En *Data-to-Text* en particulier, l'enjeu est crucial : la tâche se définit comme une sorte de traduction depuis des données complexes, parfois structurées, souvent numériques vers des descriptions textuelles plus compréhensibles par les humains. Ces données sont généralement partiellement ou complètement numériques : séries temporelles provenant de capteurs, tableaux de valeurs, résultats de requêtes SQL, bases de connaissances et graphes, etc. C'est un champ émergent très dynamique dans le domaine du traitement du langage naturel, (Wiseman *et al.*, 2018; Zhang *et al.*, 2019; Kale & Rastogi, 2020; Parikh *et al.*, 2020; Rebuffel *et al.*, 2022), possédant de très nombreuses applications, notamment dans les domaines scientifiques, du journalisme, de la santé, du marketing, de la finance, etc. La plupart des applications dans ce domaine nécessitent de la planification, de l'analyse numérique ou du raisonnement : cet enjeu est en passe de devenir central en *Data-to-Text* (Nie *et al.*, 2018; Herzig *et al.*, 2020).

Appliquer les techniques issues de la bibliographie présentée dans cet article sur des entrées tabulaires (ou autre) semble très pertinent. En effet, l'enjeu est alors de réaliser des opérations entre des entités, des valeurs ou sur des séries de valeurs (e.g., identification de la plus grande valeur sur une colonne du tableau, recherche du joueur ayant fait le plus de passes, etc...). L'idée de faire appel à des outils de calculs externes dans le domaine du *Data-to-Text* est à la fois originale et pertinente pour aider à la résolution de cette tâche difficile. Des chaînes de raisonnement pourraient être générées puis exécutées par des outils externes lors de l'inférence, nourissant un modèle de langue chargé de générer le texte final, comme dans la proposition émise en fin de section précédente. En utilisant des outils externes pour gérer les aspects numériques, les modèles de langue pourraient se concentrer davantage sur les aspects linguistiques et sémantiques, ce qui pourrait conduire à des améliorations significatives de la qualité des sorties produites.

4 Conclusion

Cet article s'intéresse à l'amélioration des modèles de langue en raisonnement numérique. Bien que le pré-entraînement sur des données numériques soit maintenant une approche courante, les résultats ne sont pas satisfaisants. Notre synthèse de l'état de l'art des approches d'intégration du raisonnement numérique dans les modèles de langue penche clairement en faveur des architectures hybrides mêlant génération de texte et calculateurs externes. Nous sommes convaincus du potentiel de ces modèles, à la fois en raisonnement automatique à partir d'énoncés textuels et dans le cadre des applications de *Data-to-text*. Nous travaillons au développement d'architectures capables de traduire ces problématiques en code Python, d'exécuter ces programmes puis d'incorporer les résultats retournés dans un texte pertinent.

Remerciements Ce travail est effectué dans le cadre du projet ANR PRCE ACDC (ANR-21-CE23-0007)

Références

AL-NEGHEIMISH H., MADHYASTHA P. & RUSSO A. (2021). Numerical reasoning in machine reading comprehension tasks : are we there yet ? *arXiv preprint arXiv :2109.08207*.

ANDOR D., HE L., LEE K. & PITLER E. (2019). Giving bert a calculator : Finding operations and arguments with reading comprehension. *arXiv preprint arXiv :1909.00109*.

ARACI D. (2019). Finbert : Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv :1908.10063*.

BASIRI M. E., NEMATI S., ABDAR M., CAMBRIA E. & ACHARYA U. R. (2021). Abcdm : An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, **115**, 279–294.

BLACK S., BIDERMAN S., HALLAHAN E., ANTHONY Q., GAO L., GOLDING L., HE H., LEAHY C., MCDONELL K., PHANG J. *et al.* (2022). Gpt-neox-20b : An open-source autoregressive language model. *arXiv preprint arXiv :2204.06745*.

BORSOS Z., MARINIER R., VINCENT D., KHARITONOV E., PIETQUIN O., SHARIFI M., TEBOUL O., GRANGIER D., TAGLIASACCHI M. & ZEGHIDOUR N. (2022). Audiolm : a language modeling approach to audio generation. *arXiv preprint arXiv :2209.03143*.

CHEN J., GUO H., YI K., LI B. & ELHOSEINY M. (2022). Visualgpt : Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 18030–18040.

CHEN K., XU W., CHENG X., XIAOCHUAN Z., ZHANG Y., SONG L., WANG T., QI Y. & CHU W. (2020). Question directed graph attention network for numerical reasoning over text. *arXiv preprint arXiv :2009.07448*.

CHENG Y., YE Z., WANG M. & ZHANG Q. (2019). Document classification based on convolutional neural network and hierarchical attention network. *Neural Network World*, **29**(2), 83–98.

COBBE K., KOSARAJU V., BAVARIAN M., CHEN M., JUN H., KAISER L., PLAPPERT M., TWOREK J., HILTON J., NAKANO R. *et al.* (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv :2110.14168*.

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*.

DUA D., WANG Y., DASIGI P., STANOVSKY G., SINGH S. & GARDNER M. (2019). Drop : A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv :1903.00161*.

GELMAN R. & BUTTERWORTH B. (2005). Number and language : how are they related ? *Trends in cognitive sciences*, **9**(1), 6–10.

GEVA M., GUPTA A. & BERANT J. (2020). Injecting numerical reasoning skills into language models. *arXiv preprint arXiv :2004.04487*.

HERZIG J., NOWAK P. K., MÜLLER T., PICCINNO F. & EISENSCHLOS J. (2020). TaPas : Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4320–4333, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.398.

HOSSEINI M. J., HAJISHIRZI H., ETZIONI O. & KUSHMAN N. (2014). Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), p. 523–533, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1058.

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 *International Conference on Machine Learning*, p. 2790–2799 : PMLR.

HOWARD J. & RUDER S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv :1801.06146*.

JIAO W., WANG W., HUANG J.-T., WANG X. & TU Z. (2023). Is chatgpt a good translator ? a preliminary study. *arXiv preprint arXiv :2301.08745*.

KALE M. & RASTOGI A. (2020). Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, p. 97–102, Dublin, Ireland : Association for Computational Linguistics.

KONCEL-KEDZIORSKI R., HAJISHIRZI H., SABHARWAL A., ETZIONI O. & ANG S. D. (2015). Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, **3**, 585–597. DOI : 10.1162/tacl_a_00160.

KONCEL-KEDZIORSKI R., ROY S., AMINI A., KUSHMAN N. & HAJISHIRZI H. (2016). MAWPS : A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1152–1157, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1136.

LITHNER J. (2000). Mathematical reasoning in task solving. *Educational studies in mathematics*, p. 165–190.

LUND B. D. & WANG T. (2023). Chatting about chatgpt : how may ai and gpt impact academia and libraries ? *Library Hi Tech News*.

LUONG M.-T., PHAM H. & MANNING C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv :1508.04025*.

LYU Q., HAVALDAR S., STEIN A., ZHANG L., RAO D., WONG E., APIDIANAKI M. & CALLISON-BURCH C. (2023). Faithful chain-of-thought reasoning. *arXiv preprint arXiv :2301.13379*.

MIALON G., DESSÌ R., LOMELI M., NALMPANTIS C., PASUNURU R., RAILEANU R., ROZIÈRE B., SCHICK T., DWIVEDI-YU J., CELIKYILMAZ A. *et al.* (2023). Augmented language models : a survey. *arXiv preprint arXiv :2302.07842*.

MISHRA S., FINLAYSON M., LU P., TANG L., WELLECK S., BARAL C., RAJPUROHIT T., TAFJORD O., SABHARWAL A., CLARK P. *et al.* (2022). Lila : A unified benchmark for mathematical reasoning. *arXiv preprint arXiv :2210.17517*.

NIE F., WANG J., YAO J.-G., PAN R. & LIN C.-Y. (2018). Operation-guided neural networks for high fidelity data-to-text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 3879–3889, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1422.

NIJKAMP E., PANG B., HAYASHI H., TU L., WANG H., ZHOU Y., SAVARESE S. & XIONG C. (2022). Codegen : An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv :2203.13474*.

PARIKH A., WANG X., GEHRMANN S., FARUQUI M., DHINGRA B., YANG D. & DAS D. (2020). ToTTo : A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), p. 1173–1186, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.89.

PARISI A., ZHAO Y. & FIEDEL N. (2022). Talm : Tool augmented language models. *arXiv preprint arXiv :2205.12255*.

PI X., LIU Q., CHEN B., ZIYADI M., LIN Z., GAO Y., FU Q., LOU J.-G. & CHEN W. (2022). Reasoning like program executors. *arXiv preprint arXiv :2201.11473*.

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.

RAJPURKAR P., ZHANG J., LOPYREV K. & LIANG P. (2016). Squad : 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv :1606.05250*.

RAN Q., LIN Y., LI P., ZHOU J. & LIU Z. (2019). Numnet : Machine reading comprehension with numerical reasoning. *arXiv preprint arXiv :1910.06701*.

RAVICHANDER A., NAIK A., ROSE C. & HOVY E. (2019). EQUATE : A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, p. 349–361, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/K19-1033.

REBUFFEL C., ROBERTI M., SOULIER L., SCOUTHEETEN G., CANCELLIERE R. & GALLINARI P. (2022). Controlling hallucinations at word level in data-to-text generation. *Data Mining and Knowledge Discovery*, p. 1–37.

ROY S., VIEIRA T. & ROTH D. (2015). Reasoning about Quantities in Natural Language. *Transactions of the Association for Computational Linguistics*, **3**, 1–13. DOI : 10.1162/tacl_a_00118.

RUSH A. M., CHOPRA S. & WESTON J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv :1509.00685*.

SCAO T. L., FAN A., AKIKI C., PAVLICK E., ILIĆ S., HESSLOW D., CASTAGNÉ R., LUCCIONI A. S., YVON F., GALLÉ M. *et al.* (2022). Bloom : A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv :2211.05100*.

SCHICK T., DWIVEDI-YU J., DESSÌ R., RAILEANU R., LOMELI M., ZETTLEMOYER L., CANCEDDA N. & SCIALOM T. (2023). Toolformer : Language models can teach themselves to use tools. DOI : 10.48550/ARXIV.2302.04761.

STEINBERG E., JUNG K., FRIES J. A., CORBIN C. K., PFOHL S. R. & SHAH N. H. (2021). Language models are an effective representation learning technique for electronic health record data. *Journal of biomedical informatics*, **113**, 103637.

THOPPILAN R., DE FREITAS D., HALL J., SHAZEER N., KULSHRESHTHA A., CHENG H.-T., JIN A., BOS T., BAKER L., DU Y. *et al.* (2022). Lamda : Language models for dialog applications. *arXiv preprint arXiv :2201.08239*.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

WISEMAN S., SHIEBER S. & RUSH A. (2018). Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 3174–3187, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1356.

XUE L., BARUA A., CONSTANT N., AL-RFOU R., NARANG S., KALE M., ROBERTS A. & RAFFEL C. (2022). Byt5 : Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, **10**, 291–306.

YANG P.-J., CHEN Y. T., CHEN Y. & CER D. (2021). Nt5 ?! training t5 to perform numerical reasoning. *arXiv preprint arXiv :2104.07307*.

YANG Z., QI P., ZHANG S., BENGIO Y., COHEN W. W., SALAKHUTDINOV R. & MANNING C. D. (2018). Hotpotqa : A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv :1809.09600*.

YU A. W., DOHAN D., LUONG M.-T., ZHAO R., CHEN K., NOROUZI M. & LE Q. V. (2018). Qanet : Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv :1804.09541*.

ZHANG L., ZHANG S. & BALOG K. (2019). Table2vec : Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, p. 1029–1032.

ZHOU Y., BAO J., DUAN C., SUN H., LIANG J., WANG Y., ZHAO J., WU Y., HE X. & ZHAO T. (2022). Opera : Operation-pivoted discrete reasoning over text. *arXiv preprint arXiv :2204.14166*.

ZHU F., LEI W., HUANG Y., WANG C., ZHANG S., LV J., FENG F. & CHUA T.-S. (2021). TAT-QA : A question answering benchmark on a hybrid of tabular and textual content in finance. 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. 3277–3287, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.254.

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

Reconnaissance d'Entités Nommées fondée sur des Modèles de Langue Enrichis avec des Définitions de Types d'Entités

Jesús Lovón-Melgarejo[♠] Jose G. Moreno[♠] Romaric Besançon[♢]
Olivier Ferret[♢] Lynda Tamine[♠]
[♠]Université Paul Sabatier, IRIT, UMR 5505 CNRS, Toulouse, France
[♢]Université Paris-Saclay, CEA, List, Palaiseau, France

RÉSUMÉ

Des études récentes ont identifié de nouveaux défis dans la tâche de reconnaissance d'entités nommées (NER), tels que la reconnaissance d'entités complexes qui ne sont pas des syntagmes nominaux simples et/ou figurent dans des entrées textuelles courtes, avec une faible quantité d'informations contextuelles. Cet article propose une nouvelle approche qui relève ce défi, en se basant sur des modèles de langues pré-entraînés par enrichissement des définitions des types d'entités issus d'une base de connaissances. Les expériences menées dans le cadre de la tâche MultiCoNER I de SemEval ont montré que l'approche proposée permet d'atteindre des gains de performance par rapport aux modèles de référence de la tâche.

ABSTRACT

Named Entity Recognition based on Language Models Enriched with Entity Type Definitions

Recent studies have identified new challenges in the Named Entity Recognition (NER) task, such as recognizing complex entities that are not simple noun phrases, and/or occur in short text inputs with limited context information. This paper proposes a novel approach relying on pretrained language models (PLM) that leverage entity type definitions from knowledge bases. The experiments conducted on the MultiCoNER task of SemEval showed that the proposed approach enhances the model's performance and shows consistent gains compared to the task baselines.

MOTS-CLÉS : reconnaissance d'entités nommées, entités complexes, SemEval, type d'entité.

KEYWORDS: named entity recognition (NER), complex entities, SemEval, entity type.

1 Introduction

La Reconnaissance d'Entités Nommées (NER, *Named Entity Recognition*) (Grishman & Sundheim, 1996) consiste à détecter des groupes de mots en tant qu'entités nommées dans une phrase donnée et à identifier leur type à partir d'une liste a priori de types d'entités. Selon la taille de cette liste, la tâche de NER est classée en i) *granularité grossière* lorsque la liste des types est petite (comme les noms de personnes, d'organisations et de lieux, tels que proposés dans la tâche CoNLL (Tjong Kim Sang & De Meulder, 2003)) ou ii) *granularité fine* pour une liste plus étendue (Ling & Weld, 2021). De plus, nous pouvons distinguer différentes classes d'entités nommées, qualifiées comme traditionnelles (ex. Personne, Lieu) et non traditionnelles (ex. titres d'œuvres, tels qu'un livre ou une chanson).

Des études récentes ont identifié de nouveaux défis dans la tâche de NER (Meng *et al.*, 2021), particulièrement pour les entités non traditionnelles, dites complexes (Ashwini & Choi, 2014) car très évolutives et non identifiables par des expressions nominales (ex. titres de films comme *Avatar*) ou alors présentes dans une phrase textuelle courte présentant un faible contexte (Jayarao *et al.*, 2018). Les systèmes classiques de NER ne sont pas performants dans ces cas de figure, ne permettant pas ainsi de relever ces défis (Fetahu *et al.*, 2022). Par conséquent, de nouveaux cadres d'évaluation, dont MultiCoNER dans SemEval, ont été proposés pour évaluer les performances des modèles dans ces conditions difficiles de NER (Malmasi *et al.*, 2022b). Parmi les approches proposées dans le but de relever ce défi, on trouve l'utilisation de techniques d'augmentation de données (Gan *et al.*, 2022) ou l'augmentation de la phrase d'entrée en s'appuyant sur des bases de connaissances (Wang *et al.*, 2022). Ces travaux se concentrent sur l'exploitation de représentations contextualisées de la phrase d'entrée, qui impliquent des coûts élevés pour l'encodage de représentations textuelles de longueur importante. Cependant, à notre connaissance, les travaux à ce jour ne se sont pas focalisés sur la représentation des types d'entités. Dans ce travail, nous soutenons l'idée que les représentations contextualisées des types d'entités peuvent influencer positivement les performances d'un modèle de NER, en particulier pour une taxonomie d'entités à granularité fine, où certaines classes sont généralement sous-représentées.

Cet article propose ainsi une nouvelle approche qui exploite des *définitions des types d'entités* contextualisées, riches et pertinentes, pour enrichir un modèle de langue pré-entraîné (PLM) utilisé comme une base d'un classifieur NER. Nous créons manuellement ces définitions pour plusieurs langues et proposons une architecture de modèle pour exploiter ces représentations. Notre intuition est d'associer les entités de la phrase d'entrée à une définition détaillée et contextualisée de leurs types d'entités, conduisant à un espace de représentation adapté où les entités partageant le même type auront des représentations plus proches. Notre modèle comprend deux configurations différentes pour l'entraînement et le test. Pour la configuration d'entraînement, nous avons entraîné un PLM, XLM-Roberta, qui calcule des représentations contextualisées pour la phrase et les *définitions des types d'entités* associés obtenus à partir des annotations. Ensuite, nous avons aligné et agrégé les deux types de représentations. Les représentations finales des mots (*token*) sont ensuite transmises en entrée d'une couche de type champ aléatoire conditionnel en chaîne linéaire (CRF, *Conditional Random Fields)* (Lafferty *et al.*, 2001) pour la prédiction des entités nommées. Pour la configuration de test, nous évaluons notre modèle en n'utilisant que le PLM enrichi et la couche CRF. Nous avons testé nos modèles pour quatre langues : l'anglais, l'espagnol, le français et le portugais. Nos expérimentations ont montré que l'injection de cette connaissance aide à améliorer la performance du modèle et montre un gain constant par rapport à un modèle standard entraîné pour la tâche.

2 Travaux connexes

Ces dernières années, les PLM tels que BERT (Devlin *et al.*, 2018) ou XLM-RoBERTa (Conneau *et al.*, 2020) ont démontré leur efficacité pour améliorer les performances de la reconnaissance d'entités nommées dans plusieurs cadres d'évaluation (Jayarao *et al.*, 2018; Wang *et al.*, 2022; Jayarao *et al.*, 2018). Cependant, des études récentes ont montré que ces méthodes présentent des performances limitées pour faire face aux défis du monde réel (Meng *et al.*, 2021), tels que les entrées textuelles courtes avec un contexte limité (Jayarao *et al.*, 2018) et les entités complexes, dont les nouvelles entités émergentes telles que les titres de livres, films et chansons qui sont publiés chaque semaine (Fetahu *et al.*, 2022). Par conséquent, des travaux récents ont proposé de nouveaux cadres d'évaluation

pour relever ces défis, dont la tâche MultiCoNER dans SemEval (Malmasi *et al.*, 2022a).

L'ajout de contexte pertinent à la phrase d'entrée avec des ressources lexicales externes, telles que des bases de connaissances, a contribué à créer des représentations de *tokens* améliorées qui ont eu un impact positif sur les performances de la tâche de NER (Jayarao *et al.*, 2018; Wang *et al.*, 2022). De même, des travaux antérieurs ont proposé de pré-entraîner des PLM enrichis en fusionnant des représentations avec celles issues des bases de connaissances pour améliorer les représentations de *tokens*, ce qui s'est avéré utile dans des tâches de NLP connexes (Zhang *et al.*, 2019; Peters *et al.*, 2019). Néanmoins, ces techniques sont coûteuses en termes de calcul en raison des architectures neuronales supplémentaires impliquées.

Une autre ligne de travaux a exploré l'utilisation de PLM comme bases de connaissances pour extraire (Petroni *et al.*, 2019) et injecter des faits (Talmor *et al.*, 2020) de ces ressources externes en les transformant en énoncés textuels et en appliquant la tâche de pré-entraînement du PLM à ces énoncés. Sur la base de ces approches, nous proposons une nouvelle méthode qui utilise une *définition de type d'entité* comme représentation textuelle d'un fait de base de connaissances. Nous visons à extraire puis injecter cette information pour enrichir un PLM dans le cadre d'une configuration d'entraînement peu coûteuse.

Classe	Lang.	Définition du type d'entité
Medicine-Symptom	EN	A symptom is any sensation or change in bodily function that is experienced by a patient and is associated with a particular disease. A symptom is a medical term.
	ES	Un síntoma es cualquier sensación o cambio en la función corporal que experimenta un paciente y que se asocia a una enfermedad concreta. Un síntoma es un término médico.
	FR	Un symptôme est toute sensation ou modification d'une fonction corporelle ressentie par un patient et associée à une maladie particulière. Un symptôme est un terme médical.
	PT	Um sintoma é qualquer sensação ou mudança na função corporal que é experimentada por um paciente e está associada a uma determinada doença. Um sintoma é um termo médico.

TABLE 1 – Définitions des types d'entités créés pour les classes de granularité fine Symptôme en anglais (EN), espagnol (ES), français (FR) et portugais (PT).

3 Méthodologie

Cette section présente le système global de notre modèle NER ainsi que sa mise en œuvre détaillée, y compris la construction de définitions de types d'entités et l'architecture du modèle.

3.1 Motivations

Dans une tâche de NER, chaque mot de la phrase d'entrée est associé à un type d'entité appartenant à une liste de candidats prédéfinis. Ces candidats sont généralement associés à des catégories d'entités

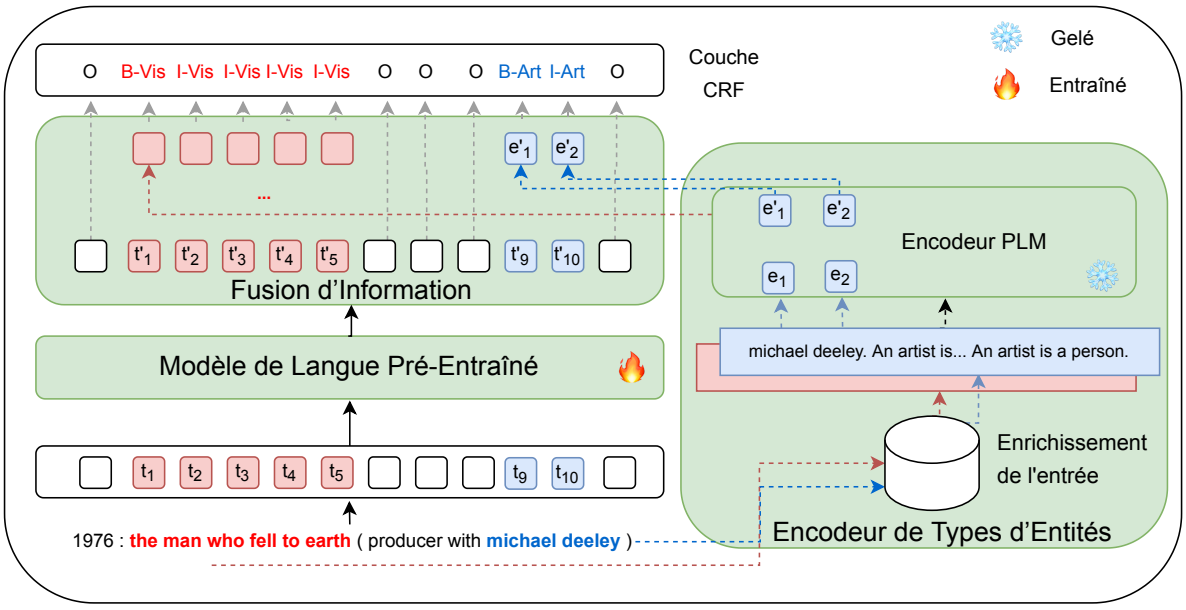


FIGURE 1 – Illustration de l'architecture du modèle.

bien définies. Cependant, quand le nombre de candidats augmente, les différences entre les catégories deviennent moins importantes, ce qui rend plus difficile leur distinction. Par exemple, il est plus facile de distinguer une entité nommée catégorisée comme *Personne* parmi une petite liste de trois autres types (ex. *Lieu*, *Organisation*, *Alimentation*) que de distinguer une entité étiquetée comme *Maladie* parmi une liste de 40 types où d'autres types, tels que *Symptôme*, sont présents.

Alors que les travaux antérieurs se sont principalement concentrés sur l'amélioration de la phrase d'entrée et des caractéristiques associées, le présent travail vise à intégrer des caractéristiques essentielles des candidats de type d'entité dans le processus d'apprentissage, ce qui pourrait profiter aux systèmes de NER. Dans ce travail, nous proposons d'utiliser des *définitions des types d'entités* (cf. Tableau 1). Une telle *définition de type d'entité* prend ici la forme d'une description textuelle du type. De plus, dans le cas des taxonomies à granularité fine, nous étendons cette définition avec une description fondée sur l'hyperonymie pour capturer la structure hiérarchique de la taxonomie. En exploitant cette information, nous visons à coder des représentations sémantiques plus riches et donc plus faciles à catégoriser comparativement à des représentations non contextualisées.

3.2 Architecture du modèle

Nous décrivons maintenant notre système NER, qui se compose de trois modules : le module *Modèle de Langue Pré-Entraîné* (PLM$_{NER}$), le module *Encodeur de Types d'Entités* et le module *Fusion d'Informations*, comme le montre la Figure 1. Notre approche consiste à adapter un PLM selon l'architecture de la Figure 1 pour améliorer les représentations du modèle pour la tâche. Mais nous nous appuyons directement sur ce seul PLM adapté pour effectuer l'inférence. Nous prenons en entrée une phrase (t) composée de n *tokens* $t = \{t_1, t_2, ..., t_n\}$ pour le module PLM$_{NER}$ et le module d'encodage des types d'entités. Cet encodeur étend d'abord les mentions d'entités de la phrase d'entrée en les concaténant avec les *définitions des types* construits manuellement, puis calcule leurs représentations. Ces représentations sont alignées avec les représentations des mots issues du module PLM$_{NER}$ puis combinées à l'aide du module de fusion d'information. Les représentations combinées

sont alors transmises à une couche CRF qui produit les prédictions des types d'entités.

Nous décrivons ci-après la nature et le rôle de chacun des modules composant le modèle proposé.

Modèle de langue pré-entraîné Ce module se réduit à son seul constituant, en l'occurrence un PLM. Étant donné une phrase d'entrée t, nous utilisons ce PLM pour calculer un ensemble de représentations contextualisées $\{t'_1, t'_2, ..., t'_n\}$, où chaque t'_i correspond à la représentation contextualisée du *token* t_i de la phrase en entrée t.

Encodeur de types d'entités À partir de l'entrée annotée de l'ensemble d'entraînement, nous identifions les mentions d'entités (représentée par une suite de *tokens* $\{e_1, e_2, .., e_m\}$) et leur type d'entité annotée. Nous étendons ensuite chaque mention d'entité en concaténant la *définition de type d'entité* correspondant (représentée par les *tokens* $\{w_1, w_2, .., w_n\}$). Enfin, le nom du type d'entité et la définition construite créent une entrée de la forme : $\{e_1, e_2, ..., w_1, ..., w_n\}$. Nous fournissons cette entrée à l'encodeur PLM de ce module pour calculer les représentations des *tokens* et ne sélectionner que les *tokens* appartenant à la mention d'entité $\{e'_1, e'_2, ...\}$.

À titre d'exemple, la Figure 1 considère la phrase d'entrée q=*"1967 : the man who fell to earth (producer with Michael Deeley)"*. Cette phrase a deux types d'entités reconnus : *Œuvre Visuelle (Vis)*, qui est un type de *Travail Créatif*, et *Artiste (Art)*, qui est un type de *Personne*, correspondant aux sous-textes *"the man who fell to earth"* et *"michael deeley"*, respectivement. L'encodeur de types d'entités génère deux phrases en concaténant les définitions des types aux entités. Plus précisément, les phrases générées sont q_1=*"the man who fell to earth. A visual work is [définition]. A visual work is a creative work."* et q_2=*"michael deeley. An artist is [définition]. An artist is a person."*. Ensuite, q_1 et q_2 sont passées dans un encodeur PLM pour obtenir les représentations correspondant aux mentions d'entités enrichies.

Ce module vise à améliorer les représentations de *tokens* en intégrant des informations contextuelles à partir des types d'entités annotés. Comme les représentations de l'encodeur PLM de ce module sont généralement informatives, nous n'entraînons pas les éléments de ce module en gelant les paramètres de ce PLM.

Fusion d'information Nous calculons une représentation agrégée en utilisant le module PLM$_{NER}$ et l'encodeur de types d'entités. Tout d'abord, pour chaque mention d'entité, nous alignons les représentations des *tokens* en sortie de l'encodeur de types avec celles en sortie du module PLM$_{NER}$. Nous ajoutons des vecteurs nuls pour les sous-*tokens* qui n'appartiennent pas aux entités. Nous effectuons ensuite une moyenne pondérée pour calculer les représentations finales :

$$(1 - b) * t'_i + b * e'_i \tag{1}$$

où *b* est un hyperparamètre avec des valeurs entre $[0, 1]$.

Enfin, les représentations agrégées sont fournies en entrée d'une couche CRF pour prédire la meilleure séquence de types d'entités parmi toutes les séquences possibles.

4 Cadre expérimental

4.1 Jeux de données

Nous avons utilisé deux jeux de données récents pour nos évaluations, les ensembles MultiCoNER I (Malmasi *et al.*, 2022a) et MultiCoNER II (Fetahu *et al.*, 2023). Les ensembles de données sont issus de trois domaines : des phrases d'encyclopédie, des questions de QA et des requêtes Web. Ces ensembles de données fournissent des phrases à faible contexte et difficiles, dans plusieurs langues, en incluant des entités complexes. La principale différence entre ces ensembles de données réside dans la taille de l'ensemble de types. L'ensemble de données MultiCoNER I fournit un ensemble de types d'entités à granularité grossière de six types, tandis que l'ensemble de données MultiCoNER II fournit une taxonomie de granularité fine avec 36 types. Chaque ensemble de données suit le format CoNLL et les annotations des types d'entités suivent un schéma BIO.

4.2 Modèles de référence

Nous avons utilisé les modèles de référence fournis avec les jeux de données, qui s'appuient sur le modèle XLM-Roberta avec une couche CRF. Nous adoptons la dénomination *XLM-RoB* dans le reste de l'article pour désigner ces modèles de référence. *XLM-RoB* est une variante multilingue du modèle RoBERTa qui a été pré-entraînée pour plus de 100 langues. *XLM-RoB* fonctionne en générant une représentation pour chaque *token*, qui est ensuite utilisée pour prédire le type du *token* à l'aide d'une couche de classification CRF. Il convient de noter que ces modèles de référence ont été affinés avec des hyperparamètres partagés pour toutes les langues [1]. Nous avons affiné notre version de XLM-RoB, sous le nom de *XLM-RoB*$_{nous}$, en utilisant la version *large* du modèle ainsi que des hyperparamètres spécifiques pour chaque langue.

Nous avons également considéré comme référence le système le plus performant pour *MultiCoNER I*, *DAMO-NLP* (Wang *et al.*, 2022). Ce dernier est fondé sur un modèle XLM-RoBERTa large et sur un extracteur de faits à partir de bases de connaissances pour ajouter du contexte pertinent à l'entrée, par exemple un paragraphe de *Wikipédia*. À l'heure actuelle, les systèmes *MultiCoNER II* ne sont pas encore disponibles, ce qui ne nous permet pas de considérer un modèle plus performant.

4.3 Modèles proposés

À l'instar de Wang *et al.* (2022), notre système utilise XLM-Roberta comme modèle de base en raison de ses performances pour cette tâche (Malmasi *et al.*, 2022c), plus précisément dans sa version *xlm-roberta-large*, disponible au niveau du *model hub* de Hugging Face. Nous avons entraîné six modèles, un pour chaque ensemble de données monolingue : anglais et espagnol pour *MultiCoNER I*, et anglais, espagnol, portugais et français pour *MultiCoNER II* [2]. Nous avons utilisé des valeurs de $b = 0{,}15$ pour l'entraînement, avec un taux d'apprentissage de 2×10^{-5} pour l'espagnol, le français et le portugais, et 1×10^{-5} avec $b = 0{,}1$ pour l'anglais. Nous avons entraîné pendant 5 époques

1. Les scores rapportés ont été extraits de https://competitions.codalab.org/competitions/36044 et https://github.com/modelscope/AdaSeq/tree/master/examples/SemEval2023_MultiCoNER_II

2. Le français et le portugais ne sont pas disponibles dans MultiCoNER I.

avec un *batch size* de 32, sur une carte graphique Nvidia RTX6000. Le temps d'entraînement était d'environ 3 heures par modèle. Nous avons utilisé une valeur de $b = 0$ pour les tests. Conformément aux travaux antérieurs (Malmasi *et al.*, 2022a), nous avons principalement rapporté le score F1, calculé pour l'ensemble complet des étiquettes.

5 Résultats

Nous avons analysé nos résultats sur les ensembles de test *MultiCoNER I* et *II*. Pour le premier, nous avons entraîné cinq modèles avec des graines aléatoires différentes (*random seeds*) et avons rapporté la moyenne de leurs scores. En revanche, pour le second, nous n'avons entraîné qu'un seul modèle en raison d'un accès limité et d'un nombre limité d'évaluations au moment de la rédaction de cet article. Les performances de notre modèle sur les deux ensembles de données et les améliorations apportées par rapport aux modèles de référence sont présentées dans le Tableau 2. Nous considérons deux modèles de référence distincts fondés sur le modèle RoBERTa : *XML-RoB* est le modèle de référence officiel de l'évaluation *MultiCoNER I* ; *XLM-Rob*$_{nous}$ est un modèle que nous avons ré-entraîné et appliqué à *MultiCoNER II* et qui constitue une référence plus robuste.

Nous avons obtenu des améliorations pour les deux jeux de données grâce à notre approche. Par rapport au modèle de référence, *XLM-RoB*, nous avons observé une amélioration de $+4{,}5$, $+6{,}3$ pour *MultiCoNER I* en anglais et en espagnol, respectivement. De même, dans les deux versions de *MultiCoNER*, nous avons constaté des améliorations entre $+0{,}3$ et $+1{,}1$ par rapport au modèle de référence plus robuste, *XLM-RoB*$_{nous}$, sauf pour la langue anglaise.

La tâche NER en espagnol a connu l'impact le plus important, avec une augmentation allant jusqu'à $+1{,}1$ points pour le score F1. Ces résultats impliquent que notre approche améliore de manière effective les représentations de modèle en utilisant des *définitions des types d'entités* dans les classes d'ensemble de données à granularité grossière et fine. Cependant, même si notre approche a produit de meilleurs scores sur l'ensemble de données *MultiCoNER II* avec la taxonomie à granularité fine (une amélioration globale de $+0{,}38$), ils étaient inférieurs à ceux obtenus sur *MultiCoNER I* (une amélioration globale de $+0{,}45$). Cette différence suggère que la difficulté liée à la taxonomie à granularité fine de la nouvelle version de l'ensemble de données pose un défi et l'ajout de définitions textuelles d'hyperonymie n'aide pas à capturer des représentations hiérarchiques riches de façon optimale.

En outre, notre modèle est nettement moins performant que le modèle *DAMO-NLP*. Une différence

	MultiCoNER I		**MultiCoNER II**			
Modèle	**EN**	**ES**	**EN**	**ES**	**FR**	**PT**
XLM-RoB$_{nous}$	65,9	62,6	62,3	66,1	64,7	65,6
Notre modèle	65,7(-0,2)	63,7(+1,1)	62,2(-0,1)	67,1(+1,0)	65,0(+0,3)	65,9(+0,3)
XLM-Rob	61,2	57,4	-	-	-	-
DAMO-NLP	91,2	89.9	-	-	-	-

TABLE 2 – Scores F1 macro-moyens obtenus pour les jeux des données MultiCoNER I et II dans les quatre langues EN, ES, FR et PT.

notable entre les approches de *DAMO-NLP* et la nôtre réside dans la modification de l'entrée lors de l'inférence. En effet, *DAMO-NLP* étend la phrase initiale, ce qui augmente considérablement le temps d'inférence pour ce modèle (10 *tokens* en moyenne à encoder dans l'entrée originale versus 218 *tokens* avec l'entrée étendue (Wang *et al.*, 2022)). L'objectif de notre modèle est d'explorer l'impact des *définitions des types d'entités*, en utilisant seulement l'entrée originelle, ce qui favorise également une inférence plus performante en termes de temps d'exécution.

Classe	**EN**	**ES**	**FR**	**PT**
OtherLOC	+3,6	+1,7	+8,6	+3,6
HumanSettl	+2,35	+2,2	+0,7	+2,3
Station	+7,0	+10,6	+8,8	+7,0
MusicalWork	+3,1	+0,5	+3,9	+3,1
WrittenWork	+0,1	+5,3	+1,5	+0,1
OtherPER	+3,9	+2,9	+3,6	+3,9
Symptom	+28,6	+1,7	+5,8	+28,6

TABLE 3 – Classes fines améliorées par notre modèle par rapport à notre modèle de base *XLM-Rob*$_{nous}$.

Dans le Tableau 3, nous montrons les classes fines de *MultiCoNER II* ayant présenté une amélioration constante. Nos résultats révèlent une amélioration substantielle allant jusqu'à $+28{,}6$ points pour le score F1 macro dans différentes langues, indiquant que notre modèle peut incorporer des informations plus pertinentes pour certains types en particulier. Néanmoins, une limitation significative de notre approche est que nous avons obtenu des scores nuls pour les types qui n'étaient pas présents dans les annotations de l'ensemble d'entraînement, ce qui montre une capacité de généralisation limitée.

Compte tenu de ces analyses, nos résultats suggèrent globalement que l'incorporation d'informations contextuelles pertinentes à propos des types d'entités dans les modèles améliore leur performance. Cependant, un travail plus conséquent est nécessaire pour dépasser les meilleurs modèles, tels que *DAMO-NLP*, qui utilisent des ressources externes pour exploiter les représentations du modèle.

6 Conclusion

Dans cet article, nous avons présenté notre approche visant à améliorer les performances de la tâche NER dans des cas de figures présentant une complexité liée au contexte limité ou multiplicité et variabilité des types d'entités. Nous avons évalué l'efficacité de l'injection d'informations contextuelles sur les définitions des types d'entités afin d'améliorer les représentations d'un PLM. Nos évaluations ont montré des améliorations par rapport aux modèles de référence, mais ont également montré ses limites, particulièrement pour des types non vus à l'entraînement. Dans les travaux futurs, nous évaluerons comment extraire de meilleures représentations de la taxonomie hiérarchique à granularité fine et adapterons cette version améliorée de XLM-RoBERTa comme modèle de base pour d'autres approches, par exemple, comme complément à d'autres modèles exploitant les représentations des phrases en entrée.

Remerciements

Ce travail a été financé par le projet ANR-19-CE23-0028 MEERQAT. Il a en outre bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2022-AD011012638R1 attribuée par GENCI.

Références

ASHWINI S. & CHOI J. D. (2014). Targetable named entity recognition in social media. *arXiv preprint arXiv :1408.0782*.

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.

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*.

FETAHU B., CHEN Z., KAR S., ROKHLENKO O. & MALMASI S. (2023). MultiCoNER v2 : a Large Multilingual dataset for Fine-grained and Noisy Named Entity Recognition.

FETAHU B., FANG A., ROKHLENKO O. & MALMASI S. (2022). Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 2777–2790, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.200.

GAN W., LIN Y., YU G., CHEN G. & YE Q. (2022). Qtrade AI at SemEval-2022 task 11 : An unified framework for multilingual NER task. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, p. 1654–1664, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.semeval-1.228.

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

JAYARAO P., JAIN C. & SRIVASTAVA A. (2018). Exploring the importance of context and embeddings in neural ner models for task-oriented dialogue systems. *arXiv preprint arXiv :1812.02370*.

LAFFERTY J., MCCALLUM A. & PEREIRA F. C. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data.

LING X. & WELD D. (2021). Fine-grained entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, **26**(1), 94–100. DOI : 10.1609/aaai.v26i1.8122.

MALMASI S., FANG A., FETAHU B., KAR S. & ROKHLENKO O. (2022a). MultiCoNER : a Large-scale Multilingual dataset for Complex Named Entity Recognition.

MALMASI S., FANG A., FETAHU B., KAR S. & ROKHLENKO O. (2022b). SemEval-2022 Task 11 : Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)* : Association for Computational Linguistics.

MALMASI S., FANG A., FETAHU B., KAR S. & ROKHLENKO O. (2022c). SemEval-2022 task 11 : Multilingual complex named entity recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, p. 1412–1437, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.semeval-1.196.

MENG T., FANG A., ROKHLENKO O. & MALMASI S. (2021). GEMNET : Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1499–1512.

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.

PETRONI F., ROCKTASCHEL T., MILLER A. H., LEWIS P., BAKHTIN A., WU Y. & RIEDEL S. (2019). Language models as knowledge bases ? In *In : Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

TALMOR A., TAFJORD O., CLARK P., GOLDBERG Y. & BERANT J. (2020). Leap-of-thought : Teaching pre-trained models to systematically reason over implicit knowledge. *Advances in Neural Information Processing Systems*, **33**, 20227–20237.

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., SHEN Y., CAI J., WANG T., WANG X., XIE P., HUANG F., LU W., ZHUANG Y., TU K., LU W. & JIANG Y. (2022). DAMO-NLP at SemEval-2022 task 11 : A knowledge-based system for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, p. 1457–1468, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.semeval-1.200.

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, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1139.

Troisième partie

Articles déjà publiés

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

Récupération de passages basée sur un graphe d'attention amélioré par des entités

Lucas Albarede[1, 2] Lorraine Goeuriot[1] Philippe Mulhem[1] Claude Le Pape-Gardeux[2] Sylvain Marié[2] Trinidad Chardin-Segui[2]
(1) Univ. Grenoble Alpes, CNRS, Grenoble INP[*], LIG, Grenoble, France
(2) Schneider Electric Industries SAS
lucas.albarede@protonmail.com, lorraine.goeuriot@imag.fr, philippe.mulhem@imag.fr

RÉSUMÉ
La recherche de passages est cruciale dans les domaines spécialisés où les documents sont longs et complexes, tels que les brevets, les documents juridiques, les rapports scientifiques, etc. Nous explorons dans cet article l'intégration d'entités et de passages dans des modèles de graphes d'attention hétérogènes dédiés à la recherche de passages. Nous utilisons les deux architectures de recherche de passages basées sur le reclassement proposées dans (Albarede *et al.*, 2022). Nous expérimentons notre proposition sur la tâche de recherche de passages TREC CAR Y3. Les résultats obtenus montrent une amélioration par rapport aux techniques de pointe et prouvent l'efficacité de l'approche. Nos expériences montrent l'importance d'utiliser des paramètres adéquats pour une telle approche.

MOTS-CLÉS : Graph Attention Networks, Conceptual Representation.

KEYWORDS: Recherche de passages, Représentation conceptuelle.

1 Introduction

Cet article est un résumé de (Albarede *et al.*, 2023), dans lequel nous proposons d'utiliser l'apprentissage par graphe hétérogène avec des bases de connaissances pour tirer parti de la recherche de passages. Il étend un travail précédent (Albarede *et al.*, 2022), en utilisant explicitement des représentations d'entités dans le processus de recherche de passages. Les réseaux neuronaux graphes basés sur l'attention, ou *GNNs*, peuvent aider à calculer les représentations sémantiques contextualisées des passages en tenant compte des interactions entre les différentes parties d'un document, et ces représentations peuvent être utilisées efficacement pour effectuer la recherche de passages.

Dans cette contribution, nous considérons les documents comme des graphes composés de passages, de sections et d'entités, et nous étudions l'utilisation de *GNNs* pour calculer des représentations contextualisées de passages améliorées par les entités, en particulier pour la tâche de recherche de passages. Les passages et les sections sont des éléments structurels définis en fonction de la structure physique du document et dont le contenu peut couvrir plusieurs sujets, tandis que les entités sont des références sémantiques qui représentent généralement un seul concept du domaine. Ces deux éléments étant de nature différente, nous étudions comment ils pourraient interagir efficacement au sein des

∗. Institute of Engineering Univ. Grenoble Alpes.

GNNs. Dans ce qui suit, nous décrivons d'abord les éléments importants des Heterogeneous Graph Attention Networks, appelés HGATs, de (Albarede *et al.*, 2022) sur lesquels nous basons ce travail. Nous nous concentrons ensuite sur la représentation des graphes de documents en tenant compte à la fois des passages et des entités. Ensuite, nous étendons les HGATs avec des représentations de passage renforcées par des entités en introduisant l'**E**ntity-**E**nhanced **HGAT** (*EE-HGAT*). Enfin, nous intégrons les représentations de passages dans un modèle de classement neuronal et réalisons des expériences sur la tâche TREC CAR Y3 Passage Retrieval.

2 Vue d'Ensemble des HGATs

Les GAT (Graph Attention Networks) sont des réseaux neuronaux multicouches qui calculent les représentations sémantiques des nœuds d'un graphe en tenant compte des informations fournies par leurs voisins (Bahdanau *et al.*, 2014). Les HGATs (Albarede *et al.*, 2022) s'appuient sur les GAT en intégrant différents types d'arêtes. Chacune de leurs couches contient : (1) Un *composant d'échantillonnage* qui définit le voisinage N_i du nœud i, incluant i lui-même ; (2) un *composant de propagation* qui calcule une représentation pour le nœud i en agrégeant la représentation de chaque nœud dans son voisinage N_i en fonction du type d'arête qui les relie. Afin de les adapter à la recherche de passages, nous intégrons les modifications suivantes : *No Out* (NO) : les liens sortants sont ignorés lors du calcul de voisinage pendant le processus d'échantillonnage ; *No Self* (NS) : les boucles sont ignorées lors du calcul de voisinage pendant le processus d'échantillonnage ; *Lambda Separator* (LB) : un coefficient spécifique λ est ajouté aux boucles et $1 - \lambda$ à toutes les autres arêtes.

3 Intégration des entités dans les HGATs

Un document est représenté sous la forme d'un graphe hétérogène où les nœuds représentent les éléments structurels (passages, sections) et les entités, et les arêtes représentent leurs relations. Nous étendons le graphique de contextualisation présenté dans (Albarede *et al.*, 2022), en ajoutant le type d'*entité* du nœud et le type d'*entity_rel* de l'arête. Nous définissons la relation entre une entité et un passage comme la mention de l'entité dans le texte du passage.

Un encodeur neuronal calcule les représentations sémantiques pour chaque nœud du graphe. Les entités sont représentées soit par leur **étiquette** (label), soit par leur **description** (desc). Nos propositions pour la *composante d'échantillonnage* de l'*EE-HGAT* visent à étudier la manière dont les entités interagissent avec les passages. *Une entité doit servir de lien entre les passages* (Yu *et al.*, 2021). Sur la base de cette hypothèse, nous étudions le *composant d'échantillonnage* de **liens**. Ce composant définit N_i le voisinage du nœud d'*entité* i tel qu'il contient chaque nœud j où $A_{ji} \neq \emptyset$ ainsi que i lui-même (A étant la matrice d'adjacence associée au graphe hétérogène). Cela permet à l'information de circuler à travers les nœuds d'entité (c'est-à-dire que deux nœuds connectés à un nœud d'entité peuvent recevoir des informations l'un de l'autre). *Une entité doit injecter ses informations dans des éléments structurels* (Dong *et al.*, 2022; Ju *et al.*, 2022; Xiong *et al.*, 2017; Das *et al.*, 2019). Sur la base de cette hypothèse, nous étudions le *composant d'échantillonnage* **injecté**. Ce composant définit N_i le voisinage du nœud d'*entité* i comme l'ensemble vide : les nœuds d'*entités* propagent leurs informations et les empêche de recevoir des informations d'autres nœuds. Les représentations des nœuds *entités* ne sont pas modifiées.

Nous présentons dans le tableau 1 trois modifications de l'architecture des *HGATs* dans nos *EE-HGATs* (cf. Section 2) : i) $BASE$ pour étudier nos propositions sans aucune modification ; ii) NO et LB, d'autres expériences (soumission en cours d'examen, non rapportée ici) ont montré de bonnes performances dans les *HGATs* ; iii) NO, NS et LB, car d'autres expériences (soumission en cours d'examen, non rapportée ici) ont montré les meilleures performances moyennes dans les *HGATs* à travers plusieurs tâches de recherche de Passages.

Représentation des entités		**Composant d'échantillonnage**		**Modifications de l'architecture**			**identifiant *EE-HGAT***
étiquette	description	lien	injecter	NO	NS	LB	
✓		✓					$label\_link_{BASE}$
✓		✓		✓		✓	$label\_link_{NO\_LB}$
✓		✓		✓	✓	✓	$label\_link_{NO\_NS\_LB}$
	✓	✓					$desc\_link_{BASE}$
	✓	✓		✓		✓	$desc\_link_{NO\_LB}$
	✓	✓		✓	✓	✓	$desc\_link_{NO\_NS\_LB}$
✓			✓				$label\_inject_{BASE}$
✓			✓	✓		✓	$label\_inject_{NO\_LB}$
✓			✓	✓	✓	✓	$label\_inject_{NO\_NS\_LB}$
	✓		✓				$desc\_inject_{BASE}$
	✓		✓	✓		✓	$desc\_inject_{NO\_LB}$
	✓		✓	✓	✓	✓	$desc\_inject_{NO\_NS\_LB}$

TABLE 1 – Les modèles *EE-HGAT* utilisés.

Afin d'effectuer la recherche de passages, nous combinons nos *EE-HGATs* avec des cadres de classement de passages. Sur la base des travaux réalisés dans (Albarede *et al.*, 2022), nous considérons deux cadres : i) Le cadre standard, dans lequel la pertinence d'un passage est estimée à l'aide d'une représentation unique contenant à la fois des informations sur le contenu et le contexte ; et ii) Le cadre d'injection tardive, dans lequel la pertinence d'un passage est estimée à l'aide d'une représentation contenant des informations sur le contenu et d'une représentation contenant uniquement des informations sur le contexte. Ces cadres utilisent un encodeur qui apprend à différencier les requêtes et les documents à l'aide d'un token spécial "[Q]" (query) et d'un token spéciel "[D]" (document) ajoutés au texte. Nous introduisons un token spécial "[E]" qui est ajouté à la forme textuelle d'une entité. Conformément à la section 2, nous associons les modèles *EE-HGAT* en utilisant les combinaisons de modifications $BASE$ et NO et LB, avec le cadre standard et avec le cadre de l'injection tardive de (Albarede *et al.*, 2022) : par exemple $desc\_inject_{NO\_NS\_LB}$ utilisé avec l'injection tardive est noté $desc\_inject_{late\_NO\_NS\_LB}$.

4 Expérimentations et résultats

Nous expérimentons nos approches sur la tâche de Recherche de Passages TREC CAR Y3 (Dietz & Foley, 2019). Un **label** d'entité est le titre du document Wikipédia correspondant, et une **description** d'entité est le premier passage du document Wikipedia correspondant. Pour représenter les documents sous forme de graphe, un nœud est créé pour chaque entité et relié aux nœuds des passages la mentionnant. Nous définissons 3 types de nœuds ($passage$, $section$, $entity$) et 6 types d'arêtes : $horizontal$, pour les $passage$ voisins et $horizontal_i$ son symétrique ; $hierarchical$ pour les liens entre un nœud $section$ et un nœud $passage$ ou entre deux nœuds $section$ et $hierarchical_i$ son symétrique ; $entity\_rel$ pour la relation entre un nœud $entity$ et un nœud $passage$ et $entity\_rel_i$ son symétrique.

Nous réalisons nos expériences en utilisant le système de RI `Pyterrier` (Macdonald & Tonellotto, 2020) et le framework `Pytorch` (Paszke *et al.*, 2019). Nous avons fixé une longueur maximale de représentation de passage de 180 tokens et une longueur maximale de représentation de requête de

Modèle de classement des passages	P@10	nDCG@20	MAP
$label\_link_{std\_BASE}$	0.156 ± 0.018	0.216 ± 0.014	0.110 ± 0.021
$label\_link_{std\_NO\_LB}$	$0.196^{il} \pm 0.017$	$0.266^{il} \pm 0.010$	$0.150^{il} \pm 0.022$
$label\_link_{late\_NO\_NS\_LB}$	0.185 ± 0.038	0.255 ± 0.051	0.151 ± 0.046
$desc\_link_{std\_BASE}$	0.176 ± 0.017	0.231 ± 0.015	0.134 ± 0.023
$desc\_link_{std\_NO\_LB}$	$0.206^{il} \pm 0.017$	$0.291^{ijkl} \pm 0.005$	$0.174^{ikl} \pm 0.019$
$desc\_link_{late\_NO\_NS\_LB}$	$0.215^{i} \pm 0.046$	$0.301^{ijlk} \pm 0.044$	$0.181^{il} \pm 0.037$
$label\_inject_{std\_BASE}$	$0.194^{il} \pm 0.019$	$0.287^{ijkl} \pm 0.011$	$0.198^{ijklm} \pm 0.021$
$label\_inject_{std\_NO\_LB}$	$0.250^{ijklmnor} \pm 0.010$	$0.377^{ijklmnor} \pm 0.009$	$0.241^{ijklmnor} \pm 0.015$
$label\_inject_{late\_NO\_NS\_LB}$	$0.255^{ijklmno} \pm 0.028$	$0.381^{ijklmnor} \pm 0.037$	$0.254^{ijklmnor} \pm 0.041$
$desc\_inject_{std\_BASE}$	$0.233^{ijklmo} \pm 0.020$	$0.316^{ijklmo} \pm 0.016$	$0.211^{ijklmno} \pm 0.021$
$desc\_inject_{std\_NO\_LB}$	$0.261^{ijklmnopqr} \pm 0.012$	$0.386^{ijklmnopqr} \pm 0.004$	$0.256^{ijklmnopr} \pm 0.017$
$desc\_inject_{late\_NO\_NS\_LB}$	$\mathbf{0.265}^{ijklmnoqr} \pm 0.027$	$\mathbf{0.401}^{ijklmnor} \pm 0.041$	$\mathbf{0.262}^{ijklmnor} \pm 0.036$

TABLE 2 – Résultats (mean $\pm$ stdev). $i, ..., r$: significativité stat. pour chaque modèle dans l'ordre décroissant (test U de Mann-Whitney, p-value =0.05). En gras : meilleur résultat par colonne.

120 tokens. Nos *EE-HGAT* sont composés de 3 couches, chaque couche ayant plusieurs fonctions d'attention avec 8 têtes avec un *dropout* égal à 0.7. La recherche de passage est effectuée en 2 phases : nous utilisons *BM25* (Robertson *et al.*, 1995) pour récupérer les 100 premiers documents et nous classons les passages de ces documents à l'aide de notre proposition. Plus de détails sur l'implémentation peuvent être trouvés dans (Albarede *et al.*, 2023).

Nous présentons les résultats de nos approches dans le tableau 2 sous la forme $mean \pm st.dev$. Nous constatons que les approches exploitant la composante d'**injection** donnent globalement de meilleurs résultats que celles exploitant la composante de **lien**. Plus précisément, nous constatons que l'utilisation d'un composant d'échantillonnage par injection améliore toujours les performances de manière significative (par ex. $label\_inject_{std\_NO\_LB}$ améliore sensiblement les performances par rapport à $label\_link_{std\_NO\_LB}$ de $+41,7\%$, $+60,6\%$ et $+27.5\%$ pour les mesures nDCG@20, MAP et P@10, respectivement). Une explication potentielle de cet effet est que le **lien** permet aux passages de recevoir des informations d'autres passages par le biais d'une relation d'entité partagée, mais des passages provenant de documents très différents peuvent mentionner la même entité. Un passage peut alors recevoir des informations trop larges ou trop éloignées de son contenu intrinsèque, entravant le calcul de sa représentation contextualisée. En nous concentrant sur le type de représentation des entités, nous constatons que les approches exploitant la représentation des **desc** donnent globalement de meilleurs résultats que celles exploitant la représentation des **étiquettes**. Ainsi, les représentations d'entités plus riches sont bénéfiques pour le calcul de la représentation des passages.

5 Conclusion

Cet article présente une proposition détaillée de définition de l'extension de l'entité de HGATs pour la recherche de passages. Il décrit également comment ces HGATs peuvent être intégrés dans une stratégie de reclassement des passages. Les expériences réalisées sur le corpus CAR Y3 montrent que les architectures et les choix d'apprentissage de graphes adéquats peuvent être stables.

Remerciements

Ce travail a été partiellement soutenu par le MIAI@Grenoble Alpes (ANR-19-P3IA-0003), ainsi que par l'Association Nationale de la Recherche et de la Technologie (ANRT).

Références

ALBAREDE L., GOEURIOT L., MULHEM P., PAPE-GARDEUX C. L., MARIÉ S. & CHARDIN-SEGUI T. (2023). Entity enhanced attention graph-based passages retrieval. In *2nd Workshop on Augmented Intelligence for Technology-Assisted Reviews Systems : Evaluation Metrics and Protocols for eDiscovery and Systematic Review Systems - ECIR Workshop*.

ALBAREDE L., MULHEM P., GOEURIOT L., LE PAPE-GARDEUX C., MARIÉ S. & CHARDIN-SEGUI T. (2022). Passage retrieval on structured documents using graph attention networks. In *Proceedings of ECIR 2022*, p. 13–21, Stavanger, Norway.

BAHDANAU D., CHO K. & BENGIO Y. (2014). Neural machine translation by jointly learning to align and translate.

DAS R., GODBOLE A., KAVARTHAPU D., GONG Z., SINGHAL A., YU M., GUO X., GAO T., ZAMANI H., ZAHEER M. & MCCALLUM A. (2019). Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, p. 113–118, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-5816.

DIETZ L. & FOLEY J. (2019). Trec car y3 : Complex answer retrieval overview. In *Proceedings of Text REtrieval Conference (TREC)*.

DONG Q., LIU Y., CHENG S., WANG S., CHENG Z., NIU S. & YIN D. (2022). Incorporating explicit knowledge in pre-trained language models for passage re-ranking. DOI : 10.48550/ARXIV.2204.11673.

JU M., YU W., ZHAO T., ZHANG C. & YE Y. (2022). Grape : Knowledge graph enhanced passage reader for open-domain question answering. DOI : 10.48550/ARXIV.2210.02933.

MACDONALD C. & TONELLOTTO N. (2020). Declarative experimentation in information retrieval using pyterrier. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, p. 4526–4533 : ACM.

PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J. & CHINTALA S. (2019). Pytorch : An imperative style, high-performance deep learning library. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ-BUC, E. FOX & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 32*, p. 8024–8035. Curran Associates, Inc.

ROBERTSON S., WALKER S., JONES S., HANCOCK-BEAULIEU M. M. & GATFORD M. (1995). Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, p. 109–126 : Gaithersburg, MD : NIST.

XIONG C., CALLAN J. & LIU T.-Y. (2017). Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, p. 763–772, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3077136.3080768.

YU D., ZHU C., FANG Y., YU W., WANG S., XU Y., REN X., YANG Y. & ZENG M. (2021). Kg-fid : Infusing knowledge graph in fusion-in-decoder for open-domain question answering. *CoRR*, **abs/2110.04330**.

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

Highlighting exact matching via marking strategies for ad hoc document ranking with pretrained contextualized language models*

Lila Boualili[1] Jose G.Moreno[1] Mohand Boughanem[1]
(1) IRIT, Université de Toulouse III, Toulouse, France
lila.boualili@irit.fr, jose.moreno@irit.fr, mohand.boughanem@irit.fr

RÉSUMÉ

Les modèles de langue pré-entraînés (MLPs) à l'instar de BERT se sont révélés remarquablement efficaces pour le classement ad hoc. Contrairement aux modèles antérieurs à BERT qui nécessitent des composants neuronaux spécialisés pour capturer les différents aspects de la pertinence entre la requête et le document, les MLPs sont uniquement basés sur des blocs de "transformers" où l'attention est le seul mécanisme utilisé pour extraire des signaux à partir des interactions entre les termes de la requête et le document. Grâce à l'attention croisée du "transformer", BERT s'est avéré être un modèle d'appariement sémantique efficace. Cependant, l'appariement exact reste un signal essentiel pour évaluer la pertinence d'un document par rapport à une requête de recherche d'informations, en dehors de l'appariement sémantique. Dans cet article, nous partons de l'hypothèse que BERT pourrait bénéficier d'indices explicites d'appariement exact pour mieux s'adapter à la tâche d'estimation de pertinence. Dans ce travail, nous explorons des stratégies d'intégration des signaux d'appariement exact en utilisant des "tokens" de marquage permettant de mettre en évidence les correspondances exactes entre les termes de la requête et ceux du document. Nous constatons que cette approche de marquage simple améliore de manière significative le modèle BERT vanille de référence. Nous démontrons empiriquement l'efficacité de notre approche par le biais d'expériences exhaustives sur trois collections standards en recherche d'information (RI). Les résultats montrent que les indices explicites de correspondance exacte transmis par le marquage sont bénéfiques pour des MLPs aussi bien BERT que pour ELECTRA. Nos résultats confirment que les indices traditionnels de RI, tels que la correspondance exacte de termes, sont toujours utiles pour les nouveaux modèles contextualisés pré-entraînés tels que BERT.

MOTS-CLÉS : Deep Learning, Modèles de Languge Pré-entraînés, Classement Ad hoc, Appariement Exact.

KEYWORDS: Deep Learning, Pre-trained Language Models, Ad hoc Ranking, Exact Matching.

∗. Article publié dans Information Retrieval Journal: Lila Boualili, Jose G. Moreno, and Mohand Boughanem. "Highlighting exact matching via marking strategies for ad hoc document ranking with pretrained contextualized language models." Inf Retrieval J 25.4 (2022): 414-460. https://doi.org/10.1007/s10791-022-09414-x

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

Vers l'évaluation continue des systèmes de recherche d'information.

Petra Galuščáková[1] Romain Deveaud[1] Gabriela Gonzalez-Saez[1] Philippe Mulhem[1] Lorraine Goeuriot[1] Florina Piroi[3] Martin Popel[4]
(1) Univ. Grenoble Alpes, CNRS, Grenoble INP[*], LIG Grenoble, France
(2) Qwant, France
(3) RSA, Autriche
(4) Charles University, Prague, République Tchèque
Philippe.Mulhem@imag.fr

RÉSUMÉ

Cet article présente le corpus de données de la campagne d'évaluation LongEval, dans le cadre de CLEF 2023. L'objectif de cette campagne est d'étudier comment les systèmes de recherche d'informations réagissent aux changements des données qu'ils traitent, en particulier les documents et les requêtes. Nous détaillons les objectifs de la tâche, le processus d'acquisition de données et les mesures d'évaluation utilisés.

ABSTRACT

Toward continuous evaluation of Web Information Retrieval.

This article provides a brief overview of the LongEval evaluation campaign's data corpus, which is part of CLEF 2023. The aim of this campaign is to investigate how information retrieval systems respond to changes in the data they process, specifically documents and queries. We detail the task objectives, data acquisition process, and evaluation metrics used in the campaign.

MOTS-CLÉS : Collection de test, Recherche sur le Web.

KEYWORDS: Test collection, Web search.

1 Introduction

(Ren *et al.*, 2022 arxiv220412755) a démontré que la qualité d'un système de Recherche d'Information (SRI) neuronal profond dépend de la cohérence entre les données d'entraînement et de test. Cependant, il n'existe pas à notre connaissance de campagne d'évaluation dédié à cette étude. La tâche de recherche d'information (RI) de la campagne LongEval 2023 vise à évaluer le comportement des systèmes de recherche d'information modernes face à l'évolution des données. Plus précisément, cette tâche vise à mieux comprendre l'impact du temps sur les systèmes de RI afin : i) d'évaluer l'efficacité des différentes approches de recherche d'information dans le temps, et ii) de proposer des modèles et systèmes de RI capables de tirer parti des ensembles d'apprentissages vieillissants, tout en minimisant la baisse des performances au fil du temps. La figure 1 présente le processus d'évaluation proposé dans le cadre de cette tâche. Au temps t, un système utilise des données d'apprentissage

∗. Institute of Engineering Univ. Grenoble Alpes.

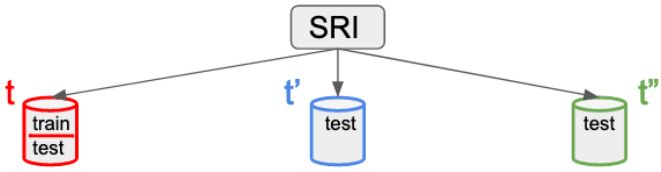


FIGURE 1 – La tâche RI de LongEval apprentissage à t, tests aux temps t, t' et t".

(documents, requêtes, évaluation de pertinence), noté *train*. A *t*, un ensemble additionnel de requêtes, de *test*, est fourni sur ces mêmes documents. L'évaluation du système sur ces requêtes de test par les organisateurs de la campagne donne la mesure de référence du système. Pour les temps ultérieurs *t'* et *t"*, qui suivent *t*, un nouveau corpus de documents et un nouvel ensemble de requêtes de test sont fournis, suivis d'une évaluation par les organisateurs de la campagne. On mesure ensuite la dégradation des résultats entre la référence de test et les temps *t'* et *t"*. En section 2, nous présentons les objectifs de la tâche de recherche d'information de la campagne LongEval. La section 3 présente les données fournies et leur acquisition. La section 4 détaille les mesures d'évaluation proposées et la section 5 des expérimentations préliminaires. Nous concluons en partie 6.

2 Principes et objectifs

La collection de la tâche de recherche d'information de Longeval s'appuie sur un large ensemble de données (un corpus de 4,2 missions de pages, 2500 de requêtes réelles, un grand nombre d'évaluations de pertinence tirées de vraies interactions avec des utilisateurs) fourni par le moteur de recherche Qwant (`https://www.qwant.com`). Elle reflète les changements de la recherche de pages Web à travers le temps, en fournissant un corpus de documents et des requêtes qui évoluent. A notre connaissance, de telles caractéristiques ne sont pas proposées dans une campagne d'évaluation, à l'échelle sur laquelle nous nous concentrons. Le paradigme de Cranfield, classique dans l'évaluation de la recherche d'information, ne prend pas en compte d'élément temporel : le corpus, les requêtes, et les évaluations de pertinence sont fixés une fois pour toute. Les collections de test Robust (Voorhees, 2006) et Twitter (Sequiera & Lin, 2017) contiennent les dates de création des documents, mais ne considèrent pas cette information en tant qu'objet d'étude. La seule collection qui intègre explicitement l'aspect temporel dans la recherche d'information ad-hoc est le récent jeu de données TREC-COVID (Voorhees *et al.*, 2021), qui propose un corpus évolutif sur le COVID. TREC-COVID contient quelques dizaines de milliers de documents et 45 requêtes au total. Dans LongEval, les échelles de données sur bien supérieures : LongEval est la seule grande collection avec des données acquises en 2022, qui a pour objectif d'évaluer les capacités des systèmes de RI modernes à se confronter à des données évolutives. De plus, le corpus de LongEval est en français et en anglais, ce qui nous différencie des collections multilingues telles que CLEF eHealth (Kelly *et al.*, 2016, 2019), dans lesquelles le français et l'anglais sont intégrées de manière limitée.

3 Les données

Nous décrivons ici le processus général d'acquisition des données issues du moteur de recherche Web Qwant, et la création des différents composants de la collection. L'acquisition globale est périodique et récurrente dans le temps afin de constituer une séquence de *sous-collections* aux temps t, t' et

t". Une sous-collection présente les caractéristiques d'une collection de test traditionnelle (requêtes, documents, jugements de pertinence), sauf qu'elle partage un ensemble commun de *sujets* (qui ne sont pas les requêtes elles-mêmes) avec les autres sous-collections. Elle est définie par :

1. L'acquisition d'un ensemble de **sujets**, sélectionnés à partir du Web et des médias sociaux. Cette acquisition est basée sur des sujets populaires - mais stables à long terme - et est effectuée une seule fois pour l'ensemble de la collection de recherche d'information LongEval.
2. La sélection de **requêtes de recherche** liées aux sujets ci-dessus, provenant des requêtes réelles émises par les utilisateurs du moteur de recherche Qwant. Elle est basée sur des intersections de chaînes de caractères entre sujets et requêtes.
3. Les **estimations de pertinence**. Nous nous appuyons ici sur deux manières de collecter ces évaluations : en utilisant implicitement des modèles de clics (Chuklin *et al.*, 2015) calculés à partir des logs de requêtes Qwant. Nous intègrerons également des évaluations manuelles qui seront collectées à la suite des runs des participants à la tâche. Étant donné que chaque sous-collection peut contenir plusieurs milliers de requêtes, nous effectuerons des évaluations explicites sur un sous-ensemble de requêtes sélectionnées manuellement.
4. L'acquisition du **corpus de documents**. Ce corpus, fourni par Qwant, est une union de : i) tous les documents Web qui ont été affichés dans la première page de résultats pour chaque requête d'une sous-collection, et ii) un échantillon aléatoire assez important de l'index Qwant. Ce protocole conduit à un corpus qui contient un mélange de documents pertinents et non pertinents. Le processus présenté gère l'évolution des pages Web, car le corpus n'est pas seulement composé d'URL, mais également du contenu des pages Web acquises à un instant.

Ces données sont acquises initialement français. Afin de permettre une participation plus ouverte à des équipes non-francophones, nous proposons une version anglaise de toutes des données (documents et requpêtes). Pour la traduction, nous avons utilisé le système français-anglais CUBBITT (Popel *et al.*, 2020), disponible à `https://lindat.cz/services/translation`. Suivant ce principe, on acquiert au temps *t*, *t'* et *t"* des corpus complets, en utilisant le même ensemble de sujets. Les requêtes de *train* au temps *t* sont sont fournies aux participants avec leurs évaluations de pertinence, et les requêtes de test sans évaluations de pertinence. Ces éléments, cf. section 4, sont utilisés par l'évaluation. Au temps *t'* et *t"*, seuls des ensembles de tests sont fournis. Les éléments du processus permettent de répondre aux objectifs de la tâche pour les raisons suivantes : a) Les corpus de documents que nous utilisons proviennent de la même source, Qwant, suivant le même principe. Ces corpus reflètent donc bien une évolution temporelle ; b) les requêtes sont tirées d'un ensemble de sujets stables dans le temps, ce qui permet d'éviter a priori des biais sur des sujets incomparables en terme de comportement humain ; c) Les évaluations de pertinence, calculées automatiquement à partir d'interactions d'utilisateurs, permettent de pouvoir gérer de grande quantités de requêtes.

4 Les mesures d'évaluation

Nous utilisons : Le nDCG, qui permet de bien prendre en compte l'importance des positions dans la liste de réponses ainsi que des mesures de pertinence multivaluées. ERR (Chapelle *et al.*, 2009) : nos estimations de pertinence étant calculées à base de Click Models, nous pouvons également considérer la métrique Expected Reciprocal Rank qui suit un modèle de navigation de l'utilisateur similaire à celui de la nDCG tout en utilisant des probabilités d'attractivité des documents. RnD(t,t') et RnD(t,t'') : la chute relative du nDCG. Ces valeurs sont égales à la différence entre le nDCG sur

les données de test au temps t et sur les ensembles de tests à t' et t''. Elles quantifient la robustesse du système évalué par rapport aux évolutions des corpus de documents et des requêtes. Avec ces mesures, un SRI de bonne qualité et répondant bien à l'évolution obtiendra une valeur de nDCG et de ERR élevée à t, t' et t'', ainsi qu'une valeur élevée pour le RnD(t,t') et RnD(t,t'').

5 Résultats sur l'ensemble d'apprentissage au temps *t*

Nous présentons dans le tableau 1 des résultats préliminaires (mesures classiques et ERR@20) sur les données d'apprentissage en français au temps t, sans apprentissage spécifique. Nous avons testé le système Terrier (Macdonald *et al.*, 2012) fournissant 1000 documents par requête, avec les paramètres par défaut (anti-dictionnaire français [1] ; troncature sur le français, *FrenchSnowballStemmer*), ainsi qu'un *reranking* de ce même BM25 par le modèle monoT5 (Nogueira *et al.*, 2020) de Castorini accessible par Pygaggle, réglé finement sur MS MARCO v1. Ces résultats nous permettent de vérifier que le corpus proposé fonctionne correctement avec des modèles classiques, tout en présentant une forte marge de progression, comme on le voit avec un *reranking* simple basé sur T5.

Système	P@10	nDCG@10	ERR@20	MAP	nDCG	Reciprocal Rank
BM25	0,1109	0,2083	0,0379	0,1767	0,3308	0,3019
reranking T5	0,1329	0,2578	0,0460	0,2175	0,3308	0,3578

TABLE 1 – Évaluation du modèle BM25 de Terrier, et reranking top 100 par T5, sur les requêtes d'apprentissage au temps *t*.

6 Conclusion

Dans cet article, nous avons décrit le corpus d'évaluation LongEval, destiné à évaluer dans quelle mesure les systèmes modernes de RI se comportent face à l'évolution des données dans le cadre de la recherche de documents sur le Web. Nous avons détaillé les étapes d'acquisition des données en françsi, leur traduction en anglais, les mesures d'évaluation et quelques résultats utilisant un modèle classiques BM25. Le corpus est accessible via le site de Longeval https://clef-longeval.github.io/.

Remerciements

Ce travail est soutenu par le projet bilatéral ANR Kodicare, subvention ANR-19-CE23-0029 de l'Agence Nationale de la Recherche française, et par le Fonds scientifique autrichien (FWF, subvention I4471-N). Ce travail a utilisé l'infrastructure de recherche LINDAT/CLARIAH-CZ (https ://lindat.cz), soutenue par le ministère de l'Éducation, de la Jeunesse et des Sports de la République tchèque (projet LM2018101 et projet LM2023062).

1. https://www.kaggle.com/datasets/rtatman/stopword-lists-for-19-languages?select=frenchST.txt

Références

CHAPELLE O., METLZER D., ZHANG Y. & GRINSPAN P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, p. 621–630, New York, NY, USA : Association for Computing Machinery.

CHUKLIN A., MARKOV I. & RIJKE M. D. (2015). Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, **7**(3), 1–115.

KELLY L., GOEURIOT L., SUOMINEN H., NÉVÉOL A., PALOTTI J. & ZUCCON G. (2016). Overview of the CLEF eHealth evaluation lab 2016. In N. FUHR, P. QUARESMA, T. GONÇALVES, B. LARSEN, K. BALOG, C. MACDONALD, L. CAPPELLATO & N. FERRO, Éds., *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, p. 255–266, Cham : Springer International Publishing.

KELLY L., SUOMINEN H., GOEURIOT L., NEVES M., KANOULAS E., LI D., AZZOPARDI L., SPIJKER R., ZUCCON G., SCELLS H. & PALOTTI J. (2019). Overview of the CLEF eHealth evaluation lab 2019. In F. CRESTANI, M. BRASCHLER, J. SAVOY, A. RAUBER, H. MÜLLER, D. E. LOSADA, G. HEINATZ BÜRKI, L. CAPPELLATO & N. FERRO, Éds., *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, p. 322–339, Cham : Springer International Publishing.

MACDONALD C., MCCREADIE R., SANTOS R. L. & OUNIS I. (2012). From puppy to maturity : Experiences in developing terrier. *Proc. of OSIR at SIGIR*, p. 60–63.

NOGUEIRA R., JIANG Z., PRADEEP R. & LIN J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 708–718, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.63.

POPEL M., TOMKOVA M., TOMEK J., ŁUKASZ KAISER, USZKOREIT J., BOJAR O. & ŽABOKRTSKÝ Z. (2020). Transforming machine translation : a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, **11**(4381), 1–15.

REN R., QU Y., LIU J., ZHAO W. X., WU Q., DING Y., WU H., WANG H. & WEN J.-R. (2022, arxiv :2204.12755). A thorough examination on zero-shot dense retrieval. DOI : 10.48550/ARXIV.2204.12755.

SEQUIERA R. & LIN J. (2017). Finally, a downloadable test collection of tweets. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, p. 1225–1228, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3077136.3080667.

VOORHEES E., ALAM T., BEDRICK S., DEMNER-FUSHMAN D., HERSH W. R., LO K., ROBERTS K., SOBOROFF I. & WANG L. L. (2021). Trec-covid : Constructing a pandemic information retrieval test collection. *SIGIR Forum*, **54**(1). DOI : 10.1145/3451964.3451965.

VOORHEES E. M. (2006). The trec 2005 robust track. In *ACM SIGIR Forum*, volume 40, p. 41–48 : ACM New York, NY, USA.

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

CoSPLADE : Adaptation d'un Modèle Neuronal Basé sur des Représentations Parcimonieuses pour la Recherche d'Information Conversationnelle

Nam Le Hai[1], Thomas Gerald[2], Thibault Formal[1,3], Jian-Yun Nie[4], Benjamin Piwowarksi[1], Laure Soulier[1,2]
(1) Sorbonne Université, CNRS, ISIR, F-75005 Paris, France
(2) Université Paris-Saclay, CNRS, SATT Paris Saclay, LISN, 91405, Orsay, France
(3) Naver Labs Europe, Meylan, France
(4) University of Montreal, Montreal, Canada
(5) Université Paris-Saclay, CNRS, LISN, 91405, Orsay, France
first.last @{sorbonne-universite.fr,lisn.fr,naverlabs.com},
nie@iro.umontreal.ca

RÉSUMÉ

La recherche conversationnelle est une tâche qui vise à retrouver des documents à partir de la question courante de l'utilisateur ainsi que l'historique complet de la conversation. La plupart des méthodes antérieures sont basées sur une approche multi-étapes reposant sur une reformulation de la question. Cette étape de reformulation est critique, car elle peut conduire à un classement sous-optimal des documents. D'autres approches ont essayé d'ordonner directement les documents, mais s'appuient pour la plupart sur un jeu de données contenant des pseudo-labels. Dans ce travail, nous proposons une technique d'apprentissage à la fois "légère" et innovante pour un modèle contextualisé d'ordonnancement basé sur SPLADE. En s'appuyant sur les représentations parcimonieuses de SPLADE, nous montrons que notre modèle, lorsqu'il est combiné avec le modèle de ré-ordonnancement T5Mono, obtient des résultats qui sont compétitifs avec ceux obtenus par les participants des campagnes d'évaluation TREC CAsT 2020 et 2021. Le code source de notre papier ECIR (Hai *et al.*, 2023) est disponible sur https://github.com/anonymous.

ABSTRACT

CoSPLADE : Contextualizing SPLADE for Conversational Information Retrieval.

Conversational search is a difficult task that aims at retrieving documents based not only on the current user query but also on the full conversation history. Most of the previous methods are built on a multi-stage ranking approach relying on query reformulation, a critical intermediate step that might lead to a sub-optimal retrieval. Other approaches have tried to use first-stage neural rankers, but are either zero-shot or rely on learning-to-rank based on a dataset with pseudo-labels. In this work, we propose an innovative lightweight learning technique to train a first-stage ranker based on SPLADE. By relying on SPLADE sparse representations, we show that, when combined with a second-stage ranker based on T5Mono, the results are competitive on the TREC CAsT 2020 and 2021 tracks. The source code of our ECIR paper (Hai *et al.*, 2023) is available at https://github.com/anonymous.

MOTS-CLÉS : Recherche d'Information, Recherche Conversationnelle, Ordonnancement Préalable.

KEYWORDS: Information Retrieval, Conversational Search, First-Stage Ranking.

1 Introduction

Avec le développement des assistants conversationnels comme Siri, Alexa ou Cortana, la Recherche d'Information (RI) Conversationnelle est devenue un domaine de recherche important (Culpepper *et al.*, 2018; Dalton *et al.*, 2019). Une recherche est effectuée au cours d'une session, à la manière d'une conversation naturelle : le besoin d'information de l'utilisateur est exprimé par une séquence d'interactions (questions ou questions puis réponses), introduisant ainsi des interdépendances complexes entre les différents "tours". Les modèles neuronaux de RI se sont avérés très performants pour cette tâche (Dalton *et al.*, 2020, 2021). Ils différent des travaux antérieurs car ils reposent sur une étape d'expansion de question basée sur l'historique (Zamani *et al.*, 2022), qui prend en compte toutes les questions passées et leurs réponses associées. Ce modèle de reformulation est généralement appris sur le jeu de données CANARD (Elgohary *et al.*, 2019), qui se compose d'une série de questions et de leurs réponses associées, ainsi que d'une question désambiguïsée – appelée *question oracle*. Cependant, s'appuyer sur une étape de reformulation est coûteux en calcul et peut s'avérer sous-optimal, comme souligné dans (Krasakis *et al.*, 2022; Lin *et al.*, 2021). (Krasakis *et al.*, 2022) utilisent ColBERT (Khattab & Zaharia, 2020) en inférence sans aucun entraînement du modèle ("zéro-shot"), en remplaçant la question par la séquence de questions, ce qui est clairement sous-optimal. (Lin *et al.*, 2021) proposent d'apprendre une représentation dense *contextualisée* de l'historique des questions, en optimisant une fonction d'ordonnancement sur un jeu de données composé de labels simulés. Le processus d'apprentissage est complexe (e.g., labels non fiables), et long.

Nous présentons ici notre participation à la campagne d'évaluation TREC CAsT (Dalton *et al.*, 2020, 2021), étendue dans une papier long à ECIR (Hai *et al.*, 2023). Nous proposons une approche faisant le lien entre l'ordonnancement préalable ("first-stage") et l'étape intermédiaire de reformulation. Notre modèle est une extension du modèle SPLADE (Formal *et al.*, 2021, 2022) basé sur l'apprentissage de représentations parcimonieuses, permettant ainsi d'extraire des mots pertinents pour la reformulation. Le processus d'apprentissage est "léger", car nous nous concentrons sur les questions et n'utilisons aucun jugement de pertinence. Nous alignons la représentation de la question contextualisée par la conversation à celle de la question désambiguïsée définie par l'*oracle*. Nous utilisons ensuite un modèle de ré-ordonnancement neuronal en tirant parti des représentations parcimonieuses pour fournir un contexte conversationnel sous la forme de mots-clés sélectionnés par SPLADE.

2 Modèle Contextualisé pour la Recherche d'Information Conversationnelle

La campagne d'évaluation TREC CAsT se concentre sur des sessions de recherche conversationnelle contenant environ 10 tours d'échanges. Chaque étape correspond à une question et la réponse canonique qui lui est associée [1], qui sert ensuite de contexte pour les questions futures. Pour chaque tour $n \leq N$, où N est le dernier tour de la conversation, nous désignons par q_n la question correspondante et a_n sa réponse canonique. Le contexte d'une question q_n au tour n correspond à l'ensemble des questions et réponses précédentes et éventuellement les réponses associées. L'objectif principal du challenge TREC CAsT est de retrouver, pour chaque question q_n et son contexte – c'est-à-dire les tours de la conversation – les passages pertinents d dans une collection de passages $\mathcal{D}$. Notre approche se base sur deux étapes d'ordonnancement.

1. Sélectionnée manuellement comme étant la réponse la plus pertinente d'un système de base.

2.1 Ordonnancement Préalable

Le modèle SPLADE (Formal *et al.*, 2021, 2022) estime le score d'un document en utilisant le produit scalaire entre la représentation parcimonieuse d'un document ($\hat{d}$) et d'une question ($\hat{q}$) : $s(\hat{q}, \hat{d}) = \hat{q} \cdot \hat{d}$. De manière similaire à (Lin *et al.*, 2021), nous supposons que la représentation du document a été bien entraînée dans le modèle original, sur la tâche standard de RI *ad-hoc*. La représentation $\hat{d}$ du document d est donc obtenue en utilisant le modèle SPLADE pré-entraîné, i.e. $\hat{d} = SPLADE(\texttt{[CLS]}\ d;\ \theta_{SPLADE})$, où θ_{SPLADE} correspond aux paramètres SPLADE originaux (Formal *et al.*, 2022) [2]. Ces paramètres ne sont pas modifiés au cours de l'apprentissage.

La représentation de la question contextualisée au tour n, désignée par $\hat{q}_{n,k}$, est obtenue par un nouveau modèle dérivé du modèle pré-entraîné SPLADE, en intégrant le contexte de la conversation, à savoir les questions précédentes et les réponses précédentes de la façon suivante :

$$\hat{q}_{n,k} = \hat{q}_n^{queries} + \hat{q}_{n,k}^{answers} \quad (1)$$

$$\hat{q}_n^{queries} = SPLADE(\texttt{[CLS]}\ q_n\ \texttt{[SEP]}\ q_1\ \texttt{[SEP]}\ \ldots\ \texttt{[SEP]}\ q_{n-1}; \theta_{queries}) \quad (2)$$

$$\hat{q}_{n,k}^{answers} = \frac{1}{k} \sum_{i=n-k}^{n-1} SPLADE(q_n\ \texttt{[SEP]}\ a_i; \theta_{answers,k}) \quad (3)$$

où $\hat{q}_n^{queries}$ encode la question actuelle dans le contexte de toutes les questions précédentes, et $\hat{q}_{n,k}^{answers}$ encode la question actuelle dans le contexte de k. Nous utilisons deux versions de SPLADE paramétrées par $\theta_{queries}$ pour l'historique complet des questions et $\theta_{answers,k}$ pour les réponses.

Entraînement. Nous proposons un entraînement basé sur deux fonctions de coût ayant pour objectif de rapprocher la représentation de la question estimée avec celle de la question oracle ainsi qu'avec le contexte de la conversation. La représentation $\hat{q}_n^*$ de la question oracle est obtenue en utilisant le modèle SPLADE original : $\hat{q}_n^* = SPLADE(q_n^*; \theta_{SPLADE})$.

La première composante de notre fonction de coût est basée sur une erreur des moindres carrés (MSE) qui compare la représentation de la question estimée avec la question oracle : $Loss_{MSE}(\hat{q}_{n,k}, \hat{q}_n^*) = MSE(\hat{q}_{n,k}, \hat{q}_n^*)$. Nous avons de plus ajouté une fonction de coût MSE asymétrique, conçue pour encourager l'expansion des termes à partir des réponses passées, ainsi qu'éviter d'introduire du bruit en restreignant les termes à ceux présents dans la question oracle q_n^* :

$$Loss_{asym}(\hat{q}_{n,k}^{answers}, \hat{q}_n^*) = \left(\max(\hat{q}_n^* - \hat{q}_{n,k}^{answers}, 0)\right)^2 \quad (4)$$

2.2 Ré-Ordonnancement

Nous effectuons le ré-ordonnancement en utilisant une approche T5Mono (Nogueira *et al.*, 2020), où nous enrichissons la question brute q_n avec des mots-clés identifiés par les représentations obtenues à l'issue de la première étape. La question enrichie q_n^+ pour le tour de conversation n est la suivante :

$$q_n^+ = q_n . Contexte : q_1\ q_2\ \ldots\ q_{n-1} . Mots - cles : w_1, w_2, ..., w_K \quad (5)$$

où les w_i sont les mots les plus importants du top-K que nous sélectionnons en exploitant le modèle d'ordonnancement préalable.

2. Qui peuvent être obtenus à partir de HuggingFace (Wolf *et al.*, 2020) : https://huggingface.co/naver/splade-cocondenser-ensembledistil

TREC CAsT 2020	Recall@1000	MAP@1000	MRR	nDCG@1000	nDCG@5	nDCG@3
TREC Participant (best)	63.3	30.2	59.3	52.6	-	45.8
TREC Participant (median)	52.1	15.1	42.2	36.4	-	30.4
TREC Participant (low)	27.9	1.0	5.9	11.1	-	2.2
CoSPLADE	82.4±2.0	26.9±1.5	58.1±2.9	54.2±1.8	41.2±2.4	44.0±2.7
TREC CAsT 2021	Recall@500	MAP@500	MRR	nDCG@500	nDCG@5	nDCG@3
TREC Participants 1 (best)	85.0	37.6	67.9	63.6	-	52.6
TREC Participants 2 (median)	36.4	17.6	53.4	33.6	-	37.7
TREC Participants 3 (low)	58.9	7.6	27.0	31.4	-	15.4
CoSPLADE	84.9±1.7	35.5±1.8	69.8±3	62.2±1.9	51.99±2.6	54.4±2.9

TABLE 1 – TREC CAsT 2020 and 2021 performances regarding participants

3 Évaluation et Résultats

Nous avons conçu le protocole d'évaluation de manière à satisfaire deux objectifs d'évaluation : *i)* Évaluer séparément l'efficacité des composantes d'ordonnancement des deux étapes de CoSPLADE ; *ii)* Comparer CoSPLADE avec les modèles des participants à TREC CAsT 2020 et 2021.

Pour entraîner notre modèle, nous avons utilisé le corpus CANARD, un ensemble de données conversationnelles axé sur la réécriture de questions basée sur le contexte. Plus précisément, le jeu de données CANARD est une liste d'historiques de conversations, chacune étant composée d'une série de questions, de réponses courtes (écrites par des humains) et de questions reformulées (contextualisées). Les ensembles d'entraînement, de développement et de test comprennent respectivement $31,538$, $3,418$ et $5,571$ questions contextuelles et reformulées.

Pour évaluer notre modèle, nous avons utilisé les ensembles de données TREC CAsT 2020 et 2021 qui comprennent respectivement 25 et 26 besoins en information ("topics") et une collection de documents composée de l'ensemble de données MS MARCO, d'une mise à jour de Wikipedia à partir du benchmark KILT (Petroni *et al.*, 2020) et de la collection Washington Post V4. Pour chaque besoin d'information, une conversation est disponible, alternant questions et réponses (passages sélectionnés manuellement dans la collection, i.e. réponses canoniques). Pour chaque question (216 et 239 au total), le jeu de données fournit sa forme réécrite manuellement ainsi qu'un ensemble d'environ 20 documents pertinents. Nous utilisons le premier pour définir une borne supérieure de comparaison.

L'analyse des différentes variantes de notre modèle ainsi que la comparaison avec des modèles de l'état de l'art et des participants TREC met en évidence les conclusions suivantes : 1) L'exploitation de l'historique des questions et des réponses permet de mieux contextualiser la question en cours. 2) Les réponses plus détaillées sont plus performantes. 3) Le coût asymétrique est bénéfique. 4) Notre modèle est capable d'obtenir des résultats comparables aux meilleurs modèles proposés par les participants TREC avec un modèle bien plus simple à entraîner et utilisant peu d'heuristiques.

4 Conclusion

Dans cet article, nous avons montré comment un modèle neuronal de RI basé sur des représentations parcimonieuses, à savoir SPLADE, pouvait être utilisé avec un processus d'apprentissage "léger" pour la RI conversationnelle. Nous avons obtenu des résultats comparables à ceux des systèmes les plus performants lors de la campagne d'évaluation TREC CAsT. Nous envisageons également d'évaluer notre approche sur d'autres jeux de données de QA conversationnelle, tels que CoQA (Reddy *et al.*, 2019), OR-ConvQA (Qu *et al.*, 2020), ou ConvMix (Christmann *et al.*, 2022).

5 Remerciements

Ce travail est financé par l'ANR JCJC SESAMS (ANR-18- CE23-0001) et l'ANR COST (ANR-18-CE23-0016).

Références

CHRISTMANN P., ROY R. S. & WEIKUM G. (2022). Conversational question answering on heterogeneous sources. In E. AMIGÓ, P. CASTELLS, J. GONZALO, B. CARTERETTE, J. S. CULPEPPER & G. KAZAI, Éds., *SIGIR '22 : The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, p. 144–154 : ACM. DOI : 10.1145/3477495.3531815.

CULPEPPER J. S., DIAZ F. & SMUCKER M. D. (2018). Research frontiers in information retrieval : Report from the third strategic workshop on information retrieval in lorne (SWIRL 2018). *SIGIR Forum*, **52**(1), 34–90. DOI : 10.1145/3274784.3274788.

DALTON J., XIONG C. & CALLAN J. (2019). TREC CAsT 2019 : The conversational assistance track overview. *arXiv*.

DALTON J., XIONG C. & CALLAN J. (2020). CAsT 2020 : The conversational assistance track overview. *arXiv*, p.1̃0.

DALTON J., XIONG C. & CALLAN J. (2021). TREC CAsT 2021 : The Conversational Assistance Track Overview. *arXiv*, p.7̃.

ELGOHARY A., PESKOV D. & BOYD-GRABER J. (2019). Can You Unpack That ? Learning to Rewrite Questions-in-Context. 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. 5918–5924, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1605.

FORMAL T., LASSANCE C., PIWOWARSKI B. & CLINCHANT S. (2022). From Distillation to Hard Negative Sampling : Making Sparse Neural IR Models More Effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, p. 2353–2359, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3477495.3531857.

FORMAL T., PIWOWARSKI B. & CLINCHANT S. (2021). SPLADE : Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, p. 2288–2292, New York, NY, USA : Association for Computing Machinery. DOI : 10/gm2tf2.

HAI N. L., GERALD T., FORMAL T., NIE J., PIWOWARSKI B. & SOULIER L. (2023). Cosplade : Contextualizing SPLADE for conversational information retrieval. In *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part I*, p. 537–552. DOI : 10.1007/978-3-031-28244-7_34.

KHATTAB O. & ZAHARIA M. (2020). ColBERT : Efficient and effective passage search via contextualized late interaction over BERT. *arXiv*.

KRASAKIS A. M., YATES A. & KANOULAS E. (2022). Zero-shot Query Contextualization for Conversational Search. In *Proceedings of the 45th International ACM SIGIR Conference on*

Research and Development in Information Retrieval, SIGIR '22, p. 1880–1884, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3477495.3531769.

LIN S.-C., YANG J.-H. & LIN J. (2021). Contextualized query embeddings for conversational search. *arXiv*.

NOGUEIRA R., JIANG Z., PRADEEP R. & LIN J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 708–718 : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.63.

PETRONI F., PIKTUS A., FAN A., LEWIS P., YAZDANI M., DE CAO N., THORNE J., JERNITE Y., KARPUKHIN V., MAILLARD J. *et al.* (2020). Kilt : a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv :2009.02252*.

QU C., YANG L., CHEN C., QIU M., CROFT W. B. & IYYER M. (2020). Open-retrieval conversational question answering. In J. X. HUANG, Y. CHANG, X. CHENG, J. KAMPS, V. MURDOCK, J. WEN & Y. LIU, Éds., *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, p. 539–548 : ACM. DOI : 10.1145/3397271.3401110.

REDDY S., CHEN D. & MANNING C. D. (2019). Coqa : A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, **7**, 249–266. DOI : 10.1162/tacl_a_00266.

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., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (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.

ZAMANI H., TRIPPAS J. R., DALTON J. & RADLINSKI F. (2022). Conversational Information Seeking. arXiv :2201.08808 [cs], DOI : 10.48550/arXiv.2201.08808.

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

The Power of Selecting Key Blocks with Local Pre-ranking for Long Document Information Retrieval

Article publié dans ACM Transactions on Information Systems 41 (3), pages 1-35, janvier 2023

Minghan Li[1] Diana Nicoleta Popa[2] Johan Chagnon[3] Yagmur Gizem Cinar[4]
Eric Gaussier[1]
(1) Université Grenoble Alpes, France
(2) Telepathy Labs, Switzerland
(3) University of Wollongong, Australia
(4) Amazon, United Kingdom

RÉSUMÉ

Les réseaux neuronaux profonds et les modèles fondés sur les transformeurs comme BERT ont envahi le domaine de la recherche d'informations (RI) ces dernières années. Leur succès est lié au mécanisme d'auto-attention qui permet de capturer les dépendances entre les mots indépendamment de leur distance. Cependant, en raison de sa complexité quadratique dans le nombre de mots, ce mécanisme ne peut être directement utilisé sur de longues séquences, ce qui ne permet pas de déployer entièrement les modèles neuronaux sur des documents longs pouvant contenir des milliers de mots. Trois stratégies standard ont été adoptées pour contourner ce problème. La première consiste à tronquer les documents longs, la deuxième à segmenter les documents longs en passages plus courts et la dernière à remplacer le module d'auto-attention par des modules d'attention parcimonieux. Dans le premier cas, des informations importantes peuvent être perdues et le jugement de pertinence n'est fondé que sur une partie de l'information contenue dans le document. Dans le deuxième cas, une architecture hiérarchique peut être adoptée pour construire une représentation du document sur la base des représentations de chaque passage. Cela dit, malgré ses résultats prometteurs, cette stratégie reste coûteuse en temps, en mémoire et en énergie. Dans le troisième cas, les contraintes de parcimonie peuvent conduire à manquer des dépendances importantes et, *in fine*, à des résultats sous-optimaux. L'approche que nous proposons est légèrement différente de ces stratégies et vise à capturer, dans les documents longs, les blocs les plus importants permettant de décider du statut, pertinent ou non, de l'ensemble du document. Elle repose sur trois étapes principales : (a) la sélection de blocs clés (c'est-à-dire susceptibles d'être pertinents) avec un pré-classement local en utilisant soit des modèles de RI classiques, soit un module d'apprentissage, (b) l'apprentissage d'une représentation conjointe des requêtes et des blocs clés à l'aide d'un modèle BERT standard, et (c) le calcul d'un score de pertinence final qui peut être considéré comme une agrégation d'informations de pertinence locale. Dans cet article, nous menons tout d'abord une analyse qui révèle que les signaux de pertinence peuvent apparaître à différents endroits dans les documents et que de tels signaux sont mieux capturés par des relations sémantiques que par des correspondances exactes. Nous examinons ensuite plusieurs méthodes pour sélectionner les blocs pertinents et montrons comment intégrer ces méthodes dans les modèles récents de RI.

MOTS-CLÉS : Modèles de langue, modèles neuronaux, recherche d'information dans les documents longs.

KEYWORDS: BERT-based language models, long-document neural information retrieval.

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

iQPP: Une Référence pour la Prédiction de Performances des Requêtes d'Images

Eduard Poesina[1] Radu Tudor Ionescu[1] Josiane Mothe[2]
(1) Department of Computer Science, University of Bucharest, 14 Academiei, Bucharest, Romania
(2) INSPE, IRIT UMR5505 CNRS, Université Toulouse Jean-Jaurès, 118 Rte de Narbonne, Toulouse, France
eduardgabriel.poe@gmail.com, raducu.ionescu@gmail.com,
josiane.mothe@irit.fr

RÉSUMÉ

La prédiction de la performance des requêtes (QPP) dans le contexte de la recherche d'images basée sur le contenu reste une tâche largement inexplorée, en particulier dans le scénario de la recherche par l'exemple, où la requête est une image. Pour stimuler les recherches dans ce domaine, nous proposons la première collection de référence. Nous proposons un ensemble de quatre jeux de données (PASCAL VOC 2012, Caltech-101, ROxford5k et RParis6k) avec les performances attendues pour chaque requête à l'aide de deux modèles de recherche d'images état de l'art. Nous proposons également de nouveaux prédicteurs pré et post-recherche. Les résultats empiriques montrent que la plupart des prédicteurs ne se généralisent pas aux différents scénarios d'évaluation. Nos expériences exhaustives indiquent que l'iQPP est une référence difficile, révélant une importante lacune dans la recherche qui doit être abordée dans les travaux futurs. Nous publions notre code et nos données [1]. Il s'agit du résumé étendu d'une publication acceptée à SIGIR 2023 (Poesina *et al.*, 2023).

ABSTRACT

iQPP : A Benchmark for Image Query Performance Prediction

Query performance prediction (QPP) in the context of content-based image retrieval remains a largely unexplored task, especially in the query-by-example scenario, where the query is an image. To stimulate research in this area, we propose the first benchmark. We propose a set of four datasets (PASCAL VOC 2012, Caltech-101, ROxford5k, and RParis6k) and estimate the ground-truth difficulty of each query using two state-of-the-art image retrieval models. We also propose new pre- and post-retrieval predictors. The empirical results show that most predictors do not generalize to different evaluation scenarios. Our extensive experiments indicate that iQPP is a challenging benchmark, revealing an important research gap that must be addressed in future work. We publish our code and data[1]. This is an extended abstract from a paper published at SIGIR 2023 (Poesina *et al.*, 2023).

MOTS-CLÉS : Systèmes d'information, Recherche d'information, prédiction de performance des requêtes, recherche d'images .

KEYWORDS: Information systems, Information retrieval, Query performance prediction, Content-based image retrieval.

1. https://github.com/Eduard6421/iQPP

1 Introduction

La prédiction des performances des requêtes (QPP) est la tâche qui consiste à estimer l'efficacité d'une recherche obtenue en réponse à une requête par un moteur de recherche, sans jugement de pertinence (Cronen-Townsend *et al.*, 2002). L'importance de cette tâche est reconnue en recherche d'information (Cronen-Townsend *et al.*, 2002; He & Ounis, 2004; Mothe & Tanguy, 2005; Hauff *et al.*, 2008, 2009; Shtok *et al.*, 2010; Cummins *et al.*, 2011; Kurland *et al.*, 2012; Cummins, 2014; Katz *et al.*, 2014; Raiber & Kurland, 2014; Roitman *et al.*, 2017; Chifu *et al.*, 2018; Mizzaro *et al.*, 2018; Roitman, 2018; Zamani *et al.*, 2018; Roy *et al.*, 2019; Arabzadeh *et al.*, 2020; Déjean *et al.*, 2020), et intéresse actuellement la communauté scientifique (Chen *et al.*, 2022; Datta *et al.*, 2022; Faggioli *et al.*, 2022; Jafarzadeh & Ensan, 2022). Cependant, dans le contexte de la recherche d'images, la prédiction de la performance des requêtes à retenu moins d'attention jusqu'ici, avec seulement quelques travaux publiés (Xing *et al.*, 2010; Li *et al.*, 2012; Nie *et al.*, 2012; Tian *et al.*, 2012; Jia *et al.*, 2014; Jia & Tian, 2015; Tian *et al.*, 2015; Pedronette & Torres, 2015; Sun *et al.*, 2018; Valem & Pedronette, 2021). Très peu de papiers ont par ailleurs considéré l'angle des requêtes par l'exemple, scénario dans lequel la requête est une image (Li *et al.*, 2012; Pedronette & Torres, 2015; Sun *et al.*, 2018; Valem & Pedronette, 2021).

Nous considérons que l'étude de la prédiction de performance est tout aussi importante pour l'image que pour le texte. Afin de développer l'intérêt de la communauté scientifique pour contexte de la recherche d'images basée sur le contenu, où les images doivent être retrouvées à partir d'une requête image, nous avons développé une collection de référence complète que nous appelons iQPP. Elle comprend quatre ensembles de données (PASCAL VOC 2012 (Everingham *et al.*, 2015), Caltech-101 (Li *et al.*, 2022), ROxford5k (Radenović *et al.*, 2018) et RParis6k (Radenović *et al.*, 2018)), deux systèmes de recherche d'images (Radenović *et al.*, 2019; Revaud *et al.*, 2019), ainsi que plusieurs prédicteurs de performance de requête pré- et post- recherche, pour lesquels nous fournissons les niveaux de performance prédits et réels pour deux mesures d'efficacité.

Les sections suivantes résume les ressources constituant la référence iQPP. Un descriptif plus complet se trouve dans la publication orginale de Poesina *et al.* (2023) ainsi que sur le GitHub `https://github.com/Eduard6421/iQPP`.

2 Ensembles d'images

Les quatre ensembles d'images sont PASCAL VOC 2012 (Everingham *et al.*, 2015), Caltech-101 (Li *et al.*, 2022), ROxford5k (Radenović *et al.*, 2018) et RParis6k (Radenović *et al.*, 2018). ROxford5k et RParis6k sont reconnus dans le cadre de la recherche d'images ; ils comprennnet chacun 70 requêtes. Nous avons par ailleurs adapté PASCAL VOC 2012 et Caltech-101 à la tâche de prédiction de performance. Nous avons créé 700 requêtes d'apprentissage et 700 de test pour chacun (Voir Table 1 et Figure 1).

3 Modèle de recherche et évaluation

Pour évaluer la difficulté d'une requête, nous avons considéré deux mesures d'efficacité de la recherche, la précision moyenne (AP) et la précision pour les k premiers résultats retrouvés (P@k).

TABLE 1 – Informations sur les jeux de données de la référence iQPP : nombre d'images, de requêtes d'entrainement et test pour chacun.

Jeux de données	#images	#requêtes d'entrainement	#requêtes de test
PASCAL VOC 2012	17,125	700	700
Caltech-101	9,146	700	700
ROxford5k	5,063	-	70
RParis6k	6,392	-	70

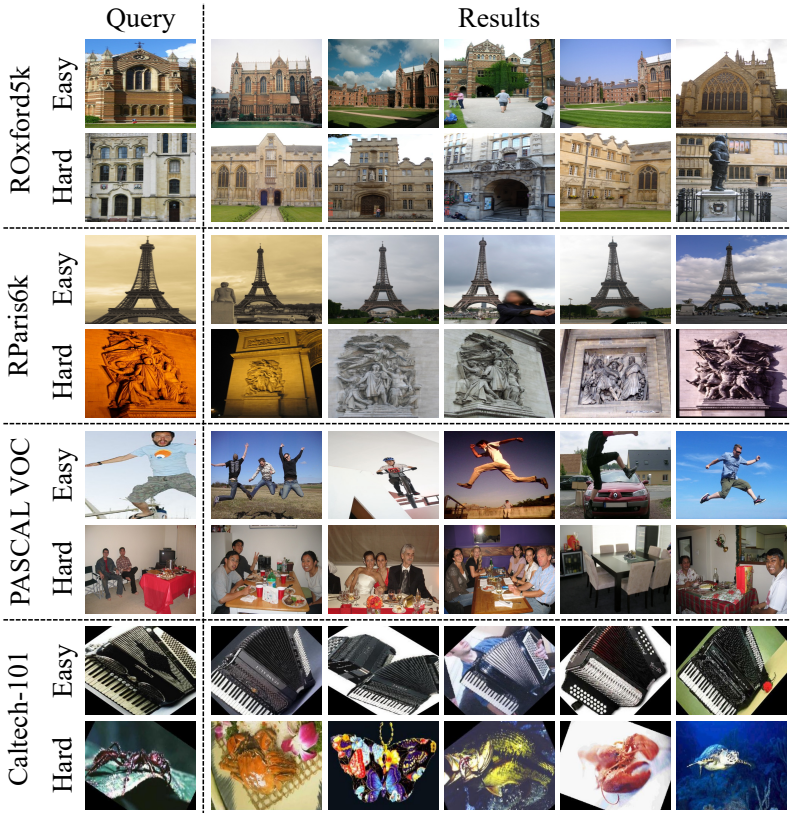


FIGURE 1 – Exemples de requêtes faciles (performance élevée) et de requêtes difficiles (faible performance) issues des 4 jeux d'images de notre référence iQPP. Pour chaque requête, nous montrons les cinq premiers résultats renvoyés par le système de Radenović et al. (Radenović *et al.*, 2019) pour illustrer les niveaux de performance des requêtes choisies. Cette image issue de (Poesina *et al.*, 2023) gagne a être vue en couleur.

Bien que la précision P@10 soit généralement utilisée dans le cadre de la prédiction de la performance des requêtes textuelles, nous avons constaté qu'un pourcentage élevé de requêtes de test dans les collections d'images (entre 29% et 82%) ont un score P@10 de 1. Pour une meilleure estimation de la difficulté de la requête, nous avons décidé d'utiliser plutôt la précision P@100.

Le premier modèle de recherche d'images que nous utilisons a été proposé par Radenović et al. (Radenović *et al.*, 2019) [2]. Il s'agit d'un modèle à base de réseau neuronal convolutif. Le second modèle

2. https://github.com/filipradenovic/cnnimageretrieval-pytorch

a été présenté par Revaud et al. (Revaud *et al.*, 2019) [3]. Le système s'appuie sur ResNet-101 (He *et al.*, 2016) pré-entraîné sur ImageNet (Russakovsky *et al.*, 2015).

Pour estimer l'efficacité d'un prédicteur, nous utilisons les coefficients de corrélation de Pearson et de Kendall τ entre les niveaux d'efficacité prédits et réels des requêtes de test, en suivant la procédure d'évaluation usuelle pour cette tâche (Yom-Tov *et al.*, 2005; Zhao *et al.*, 2008; Chifu *et al.*, 2018; Faggioli *et al.*, 2022). Nous avons utilisé un test de Student à un niveau de confiance de $0,01$ (Roitman, 2018). La figure 2 présente les résultats obtenus sur deux collections. Les prédicteurs considérés sont décrits dans la publication originale, certains sont issus de l'état de l'art (Ionescu *et al.*, 2016; Soviany *et al.*, 2021; Sun *et al.*, 2018), d'autres sont propres à cette recherche.

Type	Supervised	Method	PASCAL VOC 2012								Caltech-101							
			Radenović et al. [50]				Revaud et al. [53]				Radenović et al. [50]				Revaud et al. [53]			
			AP		P@100		AP		P@100		AP		P@100		AP		P@100	
			Pearson	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall
		Random	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Pre-retrieval		#objects / area [67]	0.02	**0.22**	0.03	**0.25**	0.02	**0.27**	0.03	**0.25**	0.01	0.08	0.01	0.04	0.04	0.06	0.03	0.04
		Image difficulty [30]	**0.25**	0.19	**0.33**	0.23	**0.32**	0.24	**0.31**	0.22	−0.01	−0.02	−0.07	−0.07	0.00	−0.02	−0.07	−0.06
		Denoising AE	0.15	0.16	0.06	0.08	0.11	0.12	0.08	0.09	0.03	0.02	0.06	0.03	0.12	0.07	0.13	0.07
		Masked AE	0.11	0.11	0.01	0.05	0.01	0.06	−0.01	0.03	−0.04	−0.04	0.01	0.00	0.03	0.02	0.09	0.05
		Class head kurtosis	0.05	0.08	0.09	0.07	0.12	0.09	0.12	0.08	0.16	0.17	0.26	0.30	0.23	0.17	0.13	0.10
		Class head dispersion	0.08	0.09	0.13	0.08	0.17	0.11	0.17	0.10	0.25	0.20	**0.48**	**0.38**	0.32	0.23	0.21	0.15
		Cluster density	0.13	0.12	0.00	0.01	−0.02	−0.04	−0.01	−0.01	0.15	0.09	0.41	0.24	−0.13	0.09	−0.03	−0.4
	✓	Fine-tuned ViT	0.04	0.02	0.20	0.10	0.17	0.06	0.14	0.05	**0.54**	**0.38**	0.27	0.15	**0.65**	**0.47**	**0.41**	**0.20**
Post-retrieval		Score Variance [13]	0.02	0.05	−0.02	0.02	0.23	0.19	0.26	0.20	0.11	0.01	0.21	0.01	0.51	0.51	0.30	0.39
	✓	Correlation CNN [68]	0.27	0.07	0.32	0.16	0.32	0.15	0.26	0.11	**0.83**	**0.65**	**0.76**	**0.51**	**0.78**	**0.60**	**0.71**	**0.50**
		Adapted query feedback	0.23	0.16	0.37	0.21	0.41	0.26	0.41	0.24	0.60	0.43	0.60	0.46	0.56	0.40	0.60	0.44
		Iterative removal	0.16	0.13	0.35	0.20	0.41	0.26	0.40	0.23	0.57	0.41	0.57	0.42	0.31	0.20	0.40	0.23
		Embedding Variance	0.29	0.20	0.33	0.21	0.43	0.22	0.37	0.20	0.28	0.20	0.49	0.28	0.26	0.18	0.49	0.26
	✓	Meta-regressor	**0.36**	**0.28**	**0.45**	**0.29**	**0.51**	**0.34**	**0.48**	**0.30**	0.71	0.53	0.72	**0.51**	0.76	0.57	0.70	0.49

FIGURE 2 – Corrélation entre les prédicteurs constituant la référence et les performances constatées des modèles de recherche - Issu de (Poesina *et al.*, 2023)

4 Conclusion

Dans cet article qui est un résumé étendu de la publication de Poesina *et al.* (2023) à la conférence SIGIR 2023, nous avons présenté la première collection pour la prédiction de performance de requêtes dans le cadre de la recherche d'images. Elle comprend quatre ensembles d'images, deux systèmes de recherche d'images et douze prédicteurs de performance de requêtes. Les résultats montrent que problème de la prédiction de performance des requêtes pour la recherche d'images est non résolu car aucun des prédicteurs n'a obtenu une performance élevée pour tous les ensembles de données.

3. https://github.com/naver/deep-image-retrieval

Références

ARABZADEH N., ZARRINKALAM F., JOVANOVIC J., AL-OBEIDAT F. & BAGHERI E. (2020). Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management*, **57**(4), 102248.

BENAMARA F., HATOUT N., MULLER P. & OZDOWSKA S., Éds. (2007). *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse. ATALA, IRIT.

CHEN X., HE B. & SUN L. (2022). Groupwise Query Performance Prediction with BERT. In *Proceedings of ECIR*, p. 64–74.

CHIFU A.-G., LAPORTE L., MOTHE J. & ULLAH M. Z. (2018). Query Performance Prediction Focused on Summarized Letor Features. In *Proceedings of SIGIR*, p. 1177–1180.

CRONEN-TOWNSEND S., ZHOU Y. & CROFT W. B. (2002). Predicting query performance. In *Proceedings of SIGIR*, p. 299–306.

CUMMINS R. (2014). Document score distribution models for query performance inference and prediction. *ACM Transactions on Information Systems*, **32**(1), 2.

CUMMINS R., JOSE J. & O'RIORDAN C. (2011). Improved query performance prediction using standard deviation. In *Proceedings of SIGIR*, p. 1089–1090.

DATTA S., MACAVANEY S., GANGULY D. & GREENE D. (2022). A'pointwise-query, listwise-document'based query performance prediction approach. In *Proceedings of SIGIR*, p. 2148–2153.

DÉJEAN S., IONESCU R. T., MOTHE J. & ULLAH M. Z. (2020). Forward and backward feature selection for query performance prediction. In *Proceedings of SAC*, p. 690–697.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

EVERINGHAM M., ESLAMI S. A., VAN GOOL L., WILLIAMS C. K., WINN J. & ZISSERMAN A. (2015). The PASCAL Visual Object Classes Challenge : A Retrospective. *International journal of computer vision*, **111**, 98–136.

FAGGIOLI G., ZENDEL O., CULPEPPER J. S., FERRO N. & SCHOLER F. (2022). sMARE : a new paradigm to evaluate and understand query performance prediction methods. *Information Retrieval Journal*, **25**(2), 94–122.

HAUFF C., AZZOPARDI L. & HIEMSTRA D. (2009). The combination and evaluation of query performance prediction methods. In *Proceedings of ECIR*, p. 301–312.

HAUFF C., HIEMSTRA D. & DE JONG F. (2008). A survey of pre-retrieval query performance predictors. In *Proceedings of CIKM*, p. 1419–1420.

HE B. & OUNIS I. (2004). Inferring query performance using pre-retrieval predictors. In *Proceedings of SPIRE*, p. 43–54.

HE K., ZHANG X., REN S. & SUN J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of CVPR*, p. 770–778.

IONESCU R., ALEXE B., LEORDEANU M., POPESCU M., PAPADOPOULOS D. P. & FERRARI V. (2016). How hard can it be ? estimating the difficulty of visual search in an image. In *Proceedings of CVPR*, p. 2157–2166.

JAFARZADEH P. & ENSAN F. (2022). A semantic approach to post-retrieval query performance prediction. *Information Processing & Management*, **59**(1), 102746.

JIA Q. & TIAN X. (2015). Query difficulty estimation via relevance prediction for image retrieval. *Signal Processing*, **110**, 232–243.

JIA Q., TIAN X. & MEI T. (2014). Query difficulty estimation via pseudo relevance feedback for image search. In *Proceedings of ICME*, p. 1–6.

KATZ G., SHTOCK A., KURLAND O., SHAPIRA B. & ROKACH L. (2014). Wikipedia-based query performance prediction. In *Proceedings of SIGIR*, p. 1235–1238.

KURLAND O., RAIBER F. & SHTOK A. (2012). Query-performance prediction and cluster ranking : Two sides of the same coin. In *Proceedings of CIKM*, p. 2459–2462.

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.

LI F.-F., ANDREETO M., RANZATO M. & PERONA P. (2022). Caltech 101. DOI : 10.22002/D1.20086.

LI Y., GENG B., YANG L., XU C. & BIAN W. (2012). Query difficulty estimation for image retrieval. *Neurocomputing*, **95**, 48–53.

MIZZARO S., MOTHE J., ROITERO K. & ULLAH M. Z. (2018). Query performance prediction and effectiveness evaluation without relevance judgments : Two sides of the same coin. In *Proceedings of SIGIR*, p. 1233–1236.

MOTHE J. & TANGUY L. (2005). Linguistic features to predict query difficulty. In *Proceedings of SIGIR*, p. 7–10.

NIE L., WANG M., ZHA Z.-J. & CHUA T.-S. (2012). Oracle in image search : a content-based approach to performance prediction. *ACM Transactions on Information Systems*, **30**(2), 1–23.

PEDRONETTE D. C. G. & TORRES R. D. S. (2015). Unsupervised effectiveness estimation for image retrieval using reciprocal rank information. In *Proceedings of SIBGRAPI*, p. 321–328.

POESINA E., IONESCU R. T. & MOTHE J. (2023). iqpp : A benchmark for image query performance prediction. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

RADENOVIĆ F., ISCEN A., TOLIAS G., AVRITHIS Y. & CHUM O. (2018). Revisiting oxford and paris : Large-scale image retrieval benchmarking. In *Proceedings of CVPR*, p. 5706–5715.

RADENOVIĆ F., TOLIAS G. & CHUM O. (2019). Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **41**(7), 1655–1668.

RAIBER F. & KURLAND O. (2014). Query-performance prediction : setting the expectations straight. In *Proceedings of SIGIR*, p. 13–22.

REVAUD J., ALMAZÁN J., REZENDE R. S. & SOUZA C. R. D. (2019). Learning with Average Precision : Training Image Retrieval with a Listwise Loss. In *Proceedings of ICCV*, p. 5107–5116.

ROITMAN H. (2018). An extended query performance prediction framework utilizing passage-level information. In *Proceedings of SIGIR*, p. 35–42.

ROITMAN H., ERERA S. & WEINER B. (2017). Robust standard deviation estimation for query performance prediction. In *Proceedings of SIGIR*, p. 245–248.

ROY D., GANGULY D., MITRA M. & JONES G. J. (2019). Estimating Gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Information Processing & Management*, **56**(3), 1026–1045.

RUSSAKOVSKY O., DENG J., SU H., KRAUSE J., SATHEESH S., MA S., HUANG Z., KARPATHY A., KHOSLA A., BERNSTEIN M. *et al.* (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211–252.

SERETAN V. & WEHRLI E. (2007). Collocation translation based on sentence alignment and parsing. In (Benamara *et al.*, 2007), p. 401–410.

SHTOK A., KURLAND O. & CARMEL D. (2010). Using statistical decision theory and relevance models for query-performance prediction. In *Proceedings of SIGIR*, p. 259–266.

SOVIANY P., IONESCU R. T., ROTA P. & SEBE N. (2021). Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, **204**, 103–166.

SUN S., ZHOU W., TIAN Q., YANG M. & LI H. (2018). Assessing image retrieval quality at the first glance. *IEEE Transactions on Image Processing*, **27**(12), 6124–6134.

TIAN X., JIA Q. & MEI T. (2015). Query difficulty estimation for image search with query reconstruction error. *IEEE Transactions on Multimedia*, **17**(1), 79–91.

TIAN X., LU Y. & YANG L. (2012). Query difficulty prediction for Web image search. *IEEE Transactions on Multimedia*, **14**(4), 951–962.

VALEM L. P. & PEDRONETTE D. C. G. (2021). A denoising convolutional neural network for self-supervised rank effectiveness estimation on image retrieval. In *Proceedings of ICMR*, p. 294–302.

XING X., ZHANG Y. & HAN M. (2010). Query difficulty prediction for contextual image retrieval. In *Proceedings of ECIR*, p. 581–585.

YOM-TOV E., FINE S., CARMEL D. & DARLOW A. (2005). Learning to estimate query difficulty : including applications to missing content detection and distributed information retrieval. In *Proceedings of SIGIR*, p. 512–519.

ZAMANI H., CROFT W. B. & CULPEPPER J. S. (2018). Neural query performance prediction using weak supervision from multiple signals. In *Proceedings of SIGIR*, p. 105–114.

ZHAO Y., SCHOLER F. & TSEGAY Y. (2008). Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proceedings of ECIR*, p. 52–64 : Springer.

Quatrième partie

Démonstration

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

XPMIR: Une bibliothèque modulaire pour l'apprentissage d'ordonnancement et les expériences de RI neuronale

Yuxuan Zong[1] Benjamin Piwowarski[1, 2]
(1) ISIR, Sorbonne Université, 4, Place Jussieu, 75005 Paris, France
(2) CNRS
yuxuan.zong@isir.upmc.fr, benjamin@piwowarski.fr

RÉSUMÉ

Ces dernières années, plusieurs librairies pour la recherche d'information (neuronale) ont été proposées. Cependant, bien qu'elles permettent de reproduire des résultats déjà publiés, il est encore très difficile de réutiliser certaines parties des chaînes de traitement d'apprentissage, comme par exemple le pré-entraînement, la stratégie d'échantillonnage ou la définition du coût dans les modèles nouvellement développés. Il est également difficile d'utiliser de nouvelles techniques d'apprentissage avec d'anciens modèles, ce qui complique l'évaluation de l'utilité des nouvelles idées pour les différents modèles de RI neuronaux. Cela ralentit l'adoption de nouvelles techniques et, par conséquent, le développement du domaine de la RI. Dans cet article, nous présentons XPMIR, une librairie Python définissant un ensemble réutilisable de composants expérimentaux. La bibliothèque contient déjà des modèles et des techniques d'indexation de pointe, et est intégrée au hub HuggingFace.

ABSTRACT

XPMIR : A Modular Library for Learning to Rank and Neural IR experiments

Those last years, several frameworks for (Neural) Information Retrieval have been proposed. However, while they allow reproducing already published results, it is still very hard to re-use some parts of the learning pipelines, as for instance the pre-training, sampling strategy, or a loss in newly developed models. It is also difficult to use new training techniques with old models, which makes it more difficult to assess the usefulness of ideas on various neural IR models. This slows the adoption of new techniques, and in turn, the development of the IR field. In this paper, we present XPMIR, a Python library defining a re-usable set of experimental components. The library already contains state-of-the-art models and indexation techniques, and is integrated with the HuggingFace hub.

MOTS-CLÉS : recherche d'information neuronale, apprentissage d'ordonnancement, cadre expérimental.

KEYWORDS: neural information retrieval, learning to rank, experimental framework.

1 Introduction

Le développement de modèles (neuronaux) de recherche d'information dépend d'une chaîne de traitement complexe (par exemple, le pré-traitement, l'entraînement, l'indexation et l'évaluation). Une série de bibliothèques (en Python) a été proposée pour faciliter la reproduction des résultats expérimentaux. Bien que ces bibliothèques soient pratiques lorsqu'il s'agit de ré-exécuter une expérience en changeant un ou des paramètres spécifiques, elles sont moins utiles lorsqu'il s'agit de créer de

nouveaux modèles.

Plus précisément, il n'est pas facile de modifier la chaîne de traitement car ces projets ne sont pas assez modulaires. Nous pensons que cela explique pourquoi les nouveaux modèles ne sont pas développés dans ces bibliothèques – empêchant ainsi l'adoption de certains *composants* dans de nouveaux modèles. Concevoir de nouveaux modèles qui tirent parti des innovations récentes prend du temps, ce qui ce qui ralentit la recherche et le partage des codes.

L'objectif de la bibliothèque XPMIR est de proposer un ensemble de *composants réutilisables* permettant de concevoir de nouvelles expériences pour la recherche d'information (neuronale), ainsi que de reproduire les anciennes. XPMIR atteint cet objectif en :

1. Adoptant des abstractions pour décrire les différentes sources de données avec des adaptateurs de jeux de données et d'échantillonneurs ;
2. Définir une manière standardisée d'apprendre, en utilisant des *hooks* (crochets) pour modifier le comportement d'apprentissage ;
3. Fournir un ensemble de composants réutilisables (échantillonneurs, modèles neuronaux, représentation textuelle, chaîne d'outils d'évaluation) ;
4. Fournir des modèles pré-entraînés sur HuggingFace. [1]

La bibliothèque est open source (licence GPLv3) et disponible sur GitHub [2]. XPMIR s'appuie sur le cadre experimaestro/datamaestro (21) qui permet (1) de standardiser l'accès aux jeu de données ; (2) de définir des composants expérimentaux modulaires (configuration et tâches) ; (3) d'exprimer des plans expérimentaux complets avec suivi automatique des dossiers (chaque dossier correspondant à un ensemble unique de paramètres expérimentaux) ; (4) de contrôler les tâches soumises (par un planificateur). Par rapport à ces librairies qui sont génériques, XPMIR apporte un ensemble de composants spécialisés *pour la recherche d'information* qui sont décrits dans la suite.

2 Travaux connexes

Tout d'abord, il existe de nombreuses bibliothèques liées à la RI. Parmi celles-ci, nous pouvons citer :

Datasets ir-datasets (13) propose une API pour accéder à de nombreux jeux de données de RI et les télécharger (s'ils sont disponibles gratuitement) ;

Indexation et recherche Il existe de nombreuses bibliothèques qui traitent de l'indexation. Pour les modèles de RI standard, nous pouvons citer PYSERINI (26), PYTERRIER (14) et Pisa (16), et pour les modèles neuronaux denses de RI, FAISS (7).

Evaluation ir-measures (13) est une bibliothèque récente qui fournit un accès direct à une grande diversité de métriques de RI.

Lorsque cela est possible, XPMIR réutilise les bibliothèques existantes en fournissant des composants qui encapsulent l'accès à ces bibliothèques, de sorte qu'elles puissent être réutilisées dans différentes expériences. Plus précisément, à ce jour, XPMIR fournit des composants pour ir-datasets, ir-measures, Pyserini, et FAISS. Une bibliothèque en RUST a également été développée pour traiter le cas des modèles neuronaux de RI parcimonieux.

1. La liste actuelle des modèles pré-entraînés est disponible sur https://huggingface.co/models?library=xpmir
2. La documentation se trouve à https://experimaestro-ir.readthedocs.io/ et le code source à https://github.com/experimaestro/experimaestro-ir

Plus en rapport avec XPMIR, diverses bibliothèques pour la RI neuronale ont été proposées ces dernières années, telles que OPENNIR [3] (11), MATCHMAKER [4] (5) ou CAPREOLUS [5] (Yates *et al.*) – ou des librairies de code plus spécifiques liés à une classe de modèles, comme par exemple pour les modèles ColBERT [6] ou SPLADE [7].

Pour OPENNIR et MATCHMAKER, les expériences sont toutes configurées par un fichier de paramètres (ou via la ligne de commande) qui permet de modifier certains aspects du processus d'apprentissage, mais pas de combiner facilement des parties de la chaîne de traitement. CAPREOLUS, qui est le plus proche de XPMIR, définit un ensemble de composants (*modules*), mais s'appuie sur des tâches prédéfinies (par exemple, apprendre et évaluer) dont certaines parties sont configurables. Toutefois, cette librairie est relativement rigide car il n'y a aucun moyen facile de modifier une chaîne de traitement.

Nous pensons que ces bibliothèques ne sont pas suffisamment modulaires. Par exemple, que se passe-t-il si nous voulons utiliser une étape de pré-entraînement MLM (Masked Language Model) spécifique tel que LexMAE (23) pour le modèle ColBERT (8) ? Que se passe-t-il si nous voulons utiliser un modèle pour générer des exemples négatifs difficiles à partir d'un nouveau modèle de RI neuronale ? Dans ces bibliothèques, il faut coder explicitement toute la "colle" entre les différentes parties, mais cela prend du temps et est source d'erreurs car les composants n'ont pas forcément été conçu pour être ré-utilisés de manière indépendante. En comparaison, la bibliothèque XPMIR a pour objectif de

— Proposer un ensemble de composants réutilisables pour apprendre et évaluer les modèles neuronaux de RI ;

— Permettre de combiner ces composants et de concevoir des expériences complexes grâce à experimaestro et datamaestro (21) ;

— Fournir un ensemble de composants d'expérience de haut niveau (par exemple ceux utilisés dans une tâche typique de ré-ordonnancement sur MS-Marco) similairement à ce qui est fait dans CAPREOLUS ;

— Fournir une intégration avec HuggingFace pour réutiliser les modèles de RI pré-entraînés.

3 XPMIR

Cette section décrit les différents composants de la bibliothèque XPMIR [8], regroupés par catégorie :

Jeux de données Ces composants permettent d'accéder à la bibliothèque ir-datasets, ainsi qu'à d'autres jeux de données spécifiques à la RI (par exemple, des triplets pour apprendre des modèles neuronaux) et à des adaptateurs qui peuvent traiter et transformer les jeux de données ;

Retrievers et Scorer Ces composants définissent comment représenter un texte, un modèle classique de RI et enfin les modèles neuronaux de RI ;

3. https://github.com/Georgetown-IR-Lab/OpenNIR
4. https://github.com/sebastian-hofstaetter/matchmaker
5. https://github.com/capreolus-ir/capreolus
6. https://github.com/stanford-futuredata/ColBERT
7. https://github.com/naver/splade
8. Ces composants correspondent soit à des configurations, soit à des tâches dans experimaestro : les configurations décrivent simplement les paramètres expérimentaux, tandis que les tâches correspondent au code réel qui peut être exécuté - par exemple, lors de l'indexation d'une collection, de l'apprentissage ou de l'évaluation d'un modèle

Apprentissage d'ordonnancement Ces composants permettent de constituer une chaîne d'entraînement ;

Évaluation Les composants permettent l'évaluation des modèles appris.

Nous illustrons certains composants par des extraits de code (modifiés) correspondant à des reproductions de deux modèles état de l'art dans XPMIR, à savoir MonoBERT (18) et SPLADE (4). MonoBERT (18) est un modèle d'encodeur joint bien établi pour la RI neuronale, qui ordonne les documents en deux étapes : sélection de candidats avec BM25 ou un modèle neuronal léger, puis ré-ordonnancement avec monoBERT. L'autre modèle qui nous sert de source pour nos exemples est SPLADE (4), un modèle d'ordonnancement qui est basé sur une représentation parcimonieuse des documents et des questions. Le code complet, qui se trouve dans la documentation [9], illustre des plans expérimentaux complexes comprenant des étapes de pré/post-traitement avec des composants d'apprentissage d'ordonnancement, ainsi que l'évaluation du modèle appris et sa comparaison avec d'autres modèles – illustrant le fait que XPMIR pourrait être utilisé pour garantir la reproductibilité d'un article de recherche en fournissant le code *complet* des expériences (y compris les modèles de base et les variations).

3.1 Jeux de données

XPMIR fournit des jeux de données au format défini par datamaestro (21), une bibliothèque qui permet d'accéder aux jeux de données de différents types de manière homogène. Grâce à datamaestro, XPMIR permet de ré-utiliser les jeux de données de ir-datasets (13), ce qui lui permet d'accéder facilement à la plupart des jeux de données de la RI. XPMIR permet également d'accéder à des jeux de données utilisés pour la distillation tels que ceux de (6). Enfin, chaque type de données – documents, questions, jugements de pertinence, triplets d'apprentissage – est associé à une interface Python permettant d'accéder aux données sous-jacentes.

XPMIR fournit également des adaptateurs [10] permettant de transformer un jeu de données. Par exemple, `RandomFold` permet échantillonner des questions d'un jeu de données de RI (et de conserver l'information sur les documents et les jugements de pertinence correspondants aux questions sélectionnées). Le code suivant montre comment échantillonner 500 questions à partir du jeu de données MS-Marco dev - en excluant les questions du jeu de données "dev small" utilisé pour évaluer les modèles dans la plupart des articles :

```
small = prepare_dataset("irds.msmarco-passage.dev.small")
dev = prepare_dataset("irds.msmarco-passage.dev")
ds_val = RandomFold(
    dataset=dev, fold=0, sizes=[500], exclude=small.topics
).submit()
```

Une autre transformation utile du jeu de données est la création d'une collection basée sur des moteurs de recherche, qui est composée de tous les documents renvoyés par un modèle de RI donné. Ceci est utile pour le calcul des mesures de validation pour les modèles neuronaux utilisés dès la première

9. https://experimaestro-ir.readthedocs.io/en/latest/papers/monobert.html pour monoBERT et https://experimaestro-ir.readthedocs.io/en/latest/papers/splade.html pour SPLADE
10. https://experimaestroir.readthedocs.io/en/latest/data/adapters.html

étape de recherche (comme SPLADE), i.e. pour rechercher de manière efficace des documents dans une grande collection.

3.2 Moteurs de recherche et les modèles neuronaux de RI

3.2.1 Représentation du texte et modèles neuronaux

De nombreux modèles s'appuient sur une certaine forme de représentation du texte. Dans XPMIR, nous distinguons trois types de représentations textuelles : (1) un `Tokenizer` qui renvoie une liste d'ID de tokens ; (2) un `TokensEncoder` qui renvoie une représentation par token ; (3) des encodeurs qui associent un vecteur à un texte (`TextEncoder`, qui peut être utilisé pour transformer une collection de documents en un index FAISS), à une paire de textes (`DualTextEncoder`, par exemple pour une paire requête/document), ou à un triplet (`TripletTextEncoder`, par exemple pour un triplet requête/document/document). Ces encodeurs sont à la base des modèles denses, des encoders joints comme MonoBERT (18) ou comme duoBERT (19). Cette représentation du texte est gérée par deux ensembles concrets de classes, celles qui correspondent à des représentations de mots comme GloVe (20) ou word2vec (17), et peuvent être utilisés pour définir des modèles pré-BERT, et ceux qui exploitent les Transformers de HuggingFace (25).

3.2.2 Retrievers and Scorers

Les `Scorers` sont chargés de calculer un score pour une question et un document donné – i.e. tous les modèles neuronaux sont des `Scorers`. Les modèles neuronaux sont organisés au sein d'une hiérarchie qui permet de factoriser autant de propriétés communes que possible. Par exemple, un `DualRepresentationScorer` est un modèle neuronal qui représente séparément les documents et les questions avant de calculer une similarité. Les modèles denses font partie de cette famille, de même que les modèles d'interaction tardive comme ColBERT (8). Un autre exemple est un `DualVectorScorer` qui s'appuie sur deux représentations vectorielles (questions et documents), et fournit des moyens d'accélérer l'apprentissage lors de l'utilisation de négatifs de batch (22). Dans la pratique, les `Scorers` sont souvent définis comme une composition d'une fonction qui calcule un score avec un modèle qui permet de représenter un texte dans un espace vectoriel. Par exemple, le modèle monoBERT peut être défini comme la composition d'un `CrossScorer`, un classificateur de représentations de couples question/document fournies par un `DualTransformerEncoder`, comme le montre le code ci-dessous [11] :

```
monobert = CrossScorer(encoder=DualTransformerEncoder(
    model_id= "bert-base-uncased", trainable=True
))
```

11. Ce code montre également comment les composants – `CrossScorer` et `DualTransformerEncoder` – peuvent être composés dans XPMIR

3.2.3 Retrievers

Les `Retrievers` permettent d'effectuer des recherches dans une *grande* collection de documents *de manière efficace*. Les modèles les plus simples sont les modèles de RI standard, comme par exemple BM25. Pour ces modèles, un `Retriever` peut être créé à partir d'un index. Pour l'instant, un adaptateur Pyserini (10) est fourni, mais d'autres comme pour PyTerrier (14) seraient faciles à mettre en œuvre. Le code suivant montre comment définir un `Retriever` pour le modèle BM25 basé sur Pyserini (10) :

```
index = IndexCollection(documents=documents).submit()
retr = AnseriniRetriever(index=index, k=50, model=BM25())
```

D'autres types d'indices sont également pris en charge pour permettre une recherche rapide à l'aide de modèles neuronaux. Les indices denses sont possibles via l'intégration de la librairie FAISS (7). Les modèles neuronaux parcimonieux produisent des indices avec une distribution de poids différente des modèles de RI standard (15). Une librairie associée a été écrite en Rust [12] et permet d'indexer une collection de documents à partir des vecteurs parcimonieux qui représentent les documents. Au niveau la recherche, les algorithmes WAND (1) et MaxScore (24) sont actuellement implémentés. À titre d'illustration, le code ci-dessous montre comment définir un `Retriever` pour le modèle SPLADE :

```
index = SparseRetrieverIndexBuilder(
    encoder=DenseDocumentEncoder(scorer=scorer), documents=documents,
).submit()
retr = SparseRetriever(index=index, topk=100,
    encoder=DenseQueryEncoder(scorer=scorer),
)
```

Enfin, pour les modèles neuronaux basés sur l'interaction, et qui ne peuvent être utilisés que pour ré-ordonner les documents, la classe `TwoStageRetriever` utilise un `Retriever` pour sélectionner un sous-ensemble de documents (ex. avec BM25) avant d'utiliser un `Scorer` pour les ré-ordonner. La définition d'un tel `Retriever` est simple :

```
retr = TwoStageRetriever(retriever=retriever, scorer=monobert)
```

3.3 Apprentissage d'ordonnancement

Le processus d'apprentissage est géré par différents composants que nous décrivons ci-dessous.

Optimisation : optimiseurs et planificateurs La partie relative à l'optimisation définit la manière d'effectuer un pas de gradient. Elle repose sur la définition d'une série d'optimiseurs, chacun étant chargé d'optimiser une partie des paramètres du modèle, ce qui est utile lorsque les paramètres

12. https://github.com/experimaestro/experimaestro-ir-rust

doivent être optimisés différemment, comme lors de l'affinage des Transformers (les couches de normalisation et les paramètres des biais ne doivent pas être inclus dans le coût de régularisation L_2). Chaque taux d'apprentissage de l'optimiseur peut être contrôlé par un planificateur – ce qui est encore une fois couramment utilisé lors de l'affinage des Transformers (25).

Le code suivant illustre comment définir l'optimiseur pour l'apprentissage de monoBERT, où le premier optimiseur utilise Adam (9) en évitant la régularisation L2 pour les biais ou les couches de normalisation (paramètres dont le nom se termine par `bias` ou contenant `LayerNorm`), tandis que le second traite tous les autres paramètres avec l'optimiseur AdamW. Cet exemple illustre de plus la flexibilité des composants exposés par XPMIR :

```
scheduler = LinearWithWarmup(num_warmup_steps=1024)
optimizers = [
    ParameterOptimizer(
        scheduler=scheduler, optimizer=Adam(),
        filter=RegexParameterFilter(includes=[r"\.bias$",
        ↪  r"\.LayerNorm\."]),
    ),
    ParameterOptimizer(
        scheduler=scheduler, optimizer=AdamW(weight_decay=1e-2),
    ),
]
```

Entraîneur and échantillonneur L'entraîneur (*Trainer*) est chargé d'effectuer une étape d'apprentissage. Il existe différents entraîneurs en fonction du type d'échantillons qu'ils peuvent traiter (ponctuelle, par paire, ou par lot). Certains entraîneurs sont conçus pour gérer la distillation, qui est essentielle pour obtenir des modèles avec des performances au niveau de l'état de l'art (**?** 3). Les données sont transmises aux entraîneurs par le biais d'échantillonneurs, qui sont chargés de fournir des échantillons de données dont le type est variable (par exemple, par points ou par paires). Enfin, des hooks peuvent être utilisés pour modifier le processus d'apprentissage, ce qui permet par exemple de calculer des coûts de régularisation.

Le code ci-dessous montre comment construire un entraîneur basé sur la distillation (utilisé pour entraîner SPLADE_DistilMSE (3)), où `sampler` est un itérateur sur les tuples composés d'une question, de deux documents et de leurs scores calculés par monoBERT :

```
distil_pairwise_trainer = DistillationPairwiseTrainer(
    batch_size=64,
    sampler=sampler,
    lossfn=MSEDifferenceLoss(),
    hooks=[FlopsRegularizer()],
)
```

Learner Enfin, la classe `Learner` est la classe qui gère l'ensemble du processus d'apprentissage. Elle s'appuie sur :

1. un modèle neuronal dont les paramètres doivent être appris (le `Scorer`);
2. un entraîneur qui spécifie comment apprendre le modèle (ex. en utilisant un coût de classification ponctuel ou par paire) et avec quelles données (à l'aide d'échantillonneurs);
3. un optimiseur qui spécifie comment effectuer la descente de gradient;
4. un ou plusieurs `Listeners` qui surveillent le processus d'apprentissage – le plus important étant celui de validation, qui permet de conserver les paramètres du modèle qui maximisent un certain nombre de métriques de validation.

En outre, le `Learner` peut être modifié à l'aide de *hooks* permettant de modifier certaines parties du processus d'apprentissage. Un exemple de *hook* permet de distribuer les modèles sur plusieurs GPU. Un autre exemple est de modifier le modèle en "gelant" certains poids – ces paramètres ne seront pas modifiés lors de l'apprentissage. Le processus d'apprentissage est divisé en époques, chaque époque étant définie par un certain nombre d'étapes d'apprentissage (c'est-à-dire par un nombre d'échantillons) effectuées par l'entraîneur. Une époque ne correspond pas à un passage complet sur le jeu de données (ce qui est nécessaire car certaines collections peuvent être très grandes, par exemple si elles sont échantillonnées à partir d'un `Retriever`). Le code suivant montre comment nous définissons un `Learner` pour monoBERT, et obtenons le modèle qui maximise la métrique `RR@10` sur un ensemble de validation :

```
learner = Learner(
    trainer=monobert_trainer, scorer=monobert_scorer,
    steps_per_epoch=100, max_epochs=1000,
    optimizers=optimizers,
    listeners=[validation],
    hooks=[DistributedHook(models=[monobert_scorer])]
)
trained = learner.submit()
best_rr10 = trained["bestval"]["RR@10"]
```

3.4 Évaluation

L'évaluation du modèle est l'étape finale des expériences de RI. Pour faciliter l'évaluation, une classe `EvaluationsCollection` référence les différents jeux de données et métriques sur lesquels l'évaluation doit être effectuée pour chaque modèle. Les métriques réelles sont calculées grâce à `ir-measures` (12). Le code ci-dessous montre un exemple d'évaluation de monoBERT – la `retriever_factory` définit comment construire un `Retriever` à partir de monoBERT (en utilisant un `TwoStageRetriever` avec `best_rr10` comme `Scorer`) en fonction de la collection de documents :

```
measures = [AP, P @ 20, nDCG, nDCG @ 10, nDCG @ 20, RR, RR @ 10]
tests = EvaluationsCollection(
    trec2019=Evaluations(
        prepare_dataset("irds.msmarco-passage.trec-dl-2019"),
        measures
    ),
    msmarco_dev=Evaluations(devsmall, measures),
```

TABLE 1 – Reproduction de monoBERT et de SPLADE

	MS MARCO dev		TREC DL 2019	
	MRR@10	nDCG@10	MRR@10	nDCG@10
monoBERT XPMIR	0.364	0.426	0.937	0.705
monoBERT (18)	0.347	-	-	-
splade-max XPMIR	0.345	0.407	0.973	0.694
splade-max (2)	0.340	-	-	0.684
splade-doc XPMIR	0.321	0.404	0.934	0.667
splade-doc (2)	0.322	-	-	0.667
splade-DistilMSE XPMIR	0.356	0.421	0.961	0.730
splade-DistilMSE (3)	0.358	-	-	0.729

```
)
tests.evaluate_retriever(retriever_factory)
```

4 Reproduction d'articles et intégration avec HuggingFace

Une partie de la bibliothèque XPMIR est dédiée à la reproduction (partielle) de certains articles de RI. Une interface (ligne de commande) est fournie pour permettre reproduire les expériences effectuées, et permettent d'automatiser le téléversement vers HuggingFace Hub, en incluant les métriques d'apprentissage et résultats de l'évaluation [13]. Deux reproductions (partielles) d'articles sont actuellement mises en œuvre dans la bibliothèque XPMIR, à savoir MonoBERT (18) et SPLADE (3). Pour la reproduction de MonoBERT, nous utilisons BERT-base comme point de départ, et pour SPLADE, DistilBERT. Dans les deux cas, nous utilisons les métriques MRR@10 et nDCG@10 et évaluons le modèle sur la recherche de passages MS-MARCO (ensemble de développement) et TREC Deep Learning 2019. Le tableau 1 montre que les résultats obtenus avec XPMIR correspondent à ceux rapportés dans les articles.

Outre la définition de composants pour la définition, l'entraînement et l'évaluation de modèles neuronaux de RI, XPMIR fournit une intégration avec le HuggingFace Hub. Cela permet de téléverser et de télécharger des modèles pré-entraînés [14]. Ces modèles peuvent être utilisés dans d'autres expériences et/ou affinés sur des jeux de données spécifiques en tirant parti de XPMIR. Le code ci-dessous montre comment créer un `Scorer` à partir du modèle dense pré-entraîné tas-balanced (6) :

```
tasb = AutoModel.load_from_hf_hub("xpmir/tas-balanced")
```

13. Un exemple avec monoBERT est disponible sur https://huggingface.co/xpmir/monobert.
14. Au moment de la rédaction, monoBERT (18), TAS-Balanced (6), ainsi que diverses versions de SPLADE (2) sont disponibles. La liste actuelle des modèles pré-entraînés est disponible sur le HuggingFace Hub : https://huggingface.co/models?library=xpmir

Ce `Scorer` peut ensuite être réutilisé dans une autre expérience – pour servir à générer de nouveaux échantillons négatifs, pour être évalué sur une nouvelle collection d'évaluation RI, ou bien pour être affiné sur d'autres jeux de données.

5 Conclusion

Dans cet article, nous avons présenté la librairie XPMIR qui permet d'entraîner et d'évaluer des modèles de RI (neuronaux). XPMIR est basé sur l'idée de décomposer le pré-traitement des ensembles de données, la représentation du texte, les modèles neuronaux de RI, l'apprentissage d'ordonnancement et l'évaluation en un ensemble de composants réutilisables – qui exploitent les bibliothèques de RI existantes dans la mesure du possible. Ces composants peuvent être composés par l'intermédiaire de la bibliothèque experimaestro (21) pour définir des plans expérimentaux complexes de RI neuronale. XPMIR fournit également des composants de haut niveau qui facilitent la description d'expériences standard (par exemple, l'entraînement d'un ré-ordonnanceur pour MS Marco).

L'intérêt d'utiliser de tels composants est qu'il est beaucoup plus facile de réutiliser certains d'entre eux pour de nouveaux modèles et de concevoir des plans expérimentaux comparant différents modèles. Avec notre structure de code actuelle et les différents composants déjà présentés, de nombreux autres articles pourraient être facilement mis en œuvre. Par exemple, nous travaillons actuellement à la reproduction de duoBERT (19) et ColBERT (8), et nous prévoyons d'inclure des procédures de pré-entraînement récentes et spécifiques à la RI comme par exemple (23) ainsi que de l'auto-distillation comme utilisée dans (3). Finalement, la librairie est open-source (licence GPLv3) et les contributions et commentaires sont les bienvenus.

Références

[1] BRODER A. Z., CARMEL D., HERSCOVICI M., SOFFER A. & ZIEN J. (2003). Efficient query evaluation using a two-level retrieval process. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, p. 426–434, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/956863.956944.

[2] FORMAL T., LASSANCE C., PIWOWARSKI B. & CLINCHANT S. (2021a). Splade v2 : Sparse lexical and expansion model for information retrieval. (arXiv :2109.10086). arXiv :2109.10086 [cs].

[3] FORMAL T., LASSANCE C., PIWOWARSKI B. & CLINCHANT S. (2022). From distillation to hard negative sampling : Making sparse neural ir models more effective. (arXiv :2205.04733). arXiv :2205.04733 [cs].

[4] FORMAL T., PIWOWARSKI B. & CLINCHANT S. (2021b). *SPLADE : Sparse Lexical and Expansion Model for First Stage Ranking*. Rapport interne. arXiv : 2107.05720.

[5] HOFSTÄTTER S. (2019). Matchmaker.

[6] HOFSTÄTTER S., LIN S.-C., YANG J.-H., LIN J. & HANBURY A. (2021). Efficiently teaching an effective dense retriever with balanced topic aware sampling. (arXiv :2104.06967). arXiv :2104.06967 [cs].

[7] JOHNSON J., DOUZE M. & JÉGOU H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, **7**(3), 535–547.

[8] KHATTAB O. & ZAHARIA M. (2020). ColBERT : Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *arXiv :2004.12832 [cs]*. arXiv : 2004.12832.

[9] KINGMA D. P. & BA J. (2017). Adam : A method for stochastic optimization.

[10] LIN J., MA X., LIN S.-C., YANG J.-H., PRADEEP R. & NOGUEIRA R. (2021). Pyserini : A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, p. 2356–2362, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3404835.3463238.

[11] MACAVANEY S. (2020). OpenNIR : A complete neural ad-hoc ranking pipeline. In *WSDM 2020*.

[12] MACAVANEY S., MACDONALD C. & OUNIS I. (2022). *Streamlining evaluation with ir-measures*, In M. HAGEN, S. VERBERNE, C. MACDONALD, C. SEIFERT, K. BALOG, K. NØRVÅG & V. SETTY, Éds., *Advances in Information Retrieval*, volume 13186 de *Lecture Notes in Computer Science*, p. 305–310. Springer International Publishing : Cham. DOI : 10.1007/978-3-030-99739-7_38.

[13] MACAVANEY S., YATES A., FELDMAN S., DOWNEY D., COHAN A. & GOHARIAN N. (2021). Simplified Data Wrangling with ir_datasets. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 2429–2436. New York, NY, USA : Association for Computing Machinery.

[14] MACDONALD C. & TONELLOTTO N. (2020). Declarative experimentation ininformation retrieval using pyterrier. In *Proceedings of ICTIR 2020*.

[15] MACKENZIE J., MALLIA A. & MOFFAT A. (2022). Accelerating Learned Sparse Indexes Via Term Impact Decomposition. In *Findings of the Association for Computational Linguistics : EMNLP 2022*.

[16] MALLIA A., SIEDLACZEK M., MACKENZIE J. & SUEL T. (2019). PISA : performant indexes and search for academia. In *Proceedings of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019, Paris, France, July 25, 2019.*, p. 50–56.

[17] MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *NIPS'14*, volume cs.CL, p. 3111–3119.

[18] NOGUEIRA R. & CHO K. (2020). Passage Re-ranking with BERT. arXiv :1901.04085 [cs], DOI : 10.48550/arXiv.1901.04085.

[19] NOGUEIRA R., YANG W., CHO K. & LIN J. (2019). Multi-Stage Document Ranking with BERT. *arXiv :1910.14424 [cs]*. ZSCC : 0000001 arXiv : 1910.14424.

[20] PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global Vectors for Word Representation.

[21] PIWOWARSKI B. (2020). Experimaestro and Datamaestro : Experiment and Dataset Managers (for IR). In *ACM SIGIR 2020*, Xian, China. ZSCC : NoCitationData[s0], DOI : 10.1145/3397271.3401410.

[22] QU Y., DING Y., LIU J., LIU K., REN R., ZHAO W. X., DONG D., WU H. & WANG H. (2021). Rocketqa : An optimized training approach to dense passage retrieval for open-domain question answering. In *In Proceedings of NAACL*.

[23] SHEN T., GENG X., TAO C., XU C., HUANG X., JIAO B., YANG L. & JIANG D. (2022). LexMAE : Lexicon-Bottlenecked Pretraining for Large-Scale Retrieval. In *ICLR* : arXiv. arXiv :2208.14754 [cs], DOI : 10.48550/arXiv.2208.14754.

[24] TURTLE H. & FLOOD J. (1995). Query evaluation : Strategies and optimizations. *Information Processing & Management*, **31**(6), 831–850. DOI : 10.1016/0306-4573(95)00020-H.

[25] 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., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). HuggingFace's Transformers : State-of-the-art Natural Language Processing. arXiv :1910.03771 [cs], DOI : 10.48550/arXiv.1910.03771.

[26] YANG P., FANG H. & LIN J. (2018). Anserini : Reproducible Ranking Baselines Using Lucene. **10**(4). DOI : 10/ggmdws.

[Yates *et al.*] YATES A., ARORA S., ZHANG X., YANG W., JOSE K. M. & LIN J. Capreolus : A Toolkit for End-to-End Neural Ad Hoc Retrieval. WSDM '20, p. 861–864. DOI : 10/ggjnkm.