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.

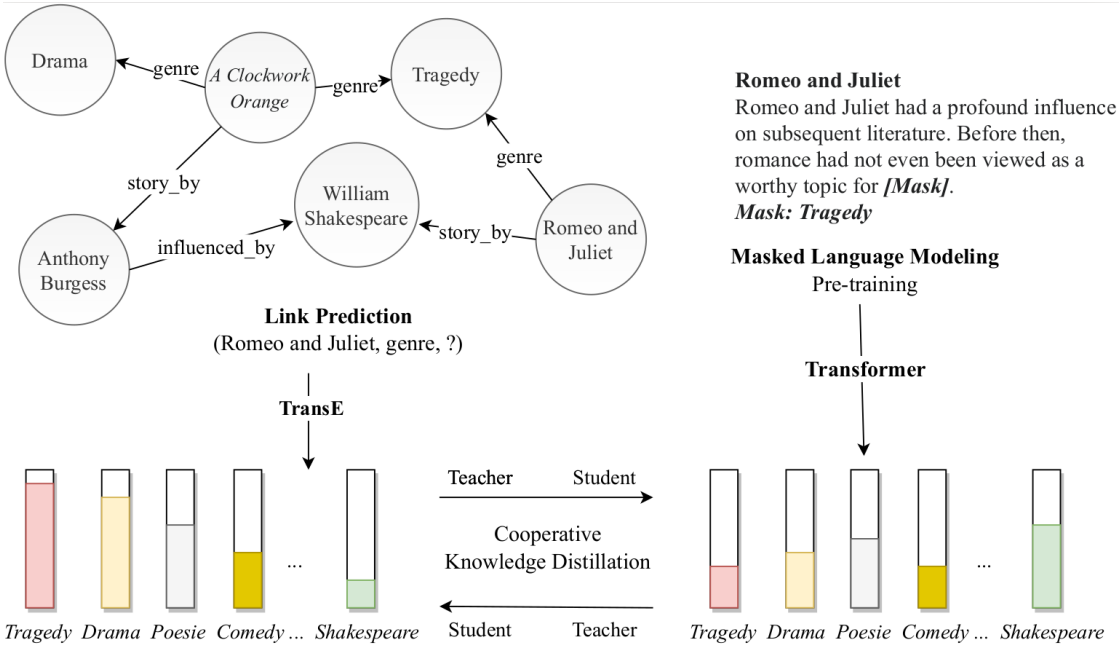

FIGURE 1 – Distillation de la connaissance entre le PLM et KB pour la tâche de complétion de connaissances.

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.

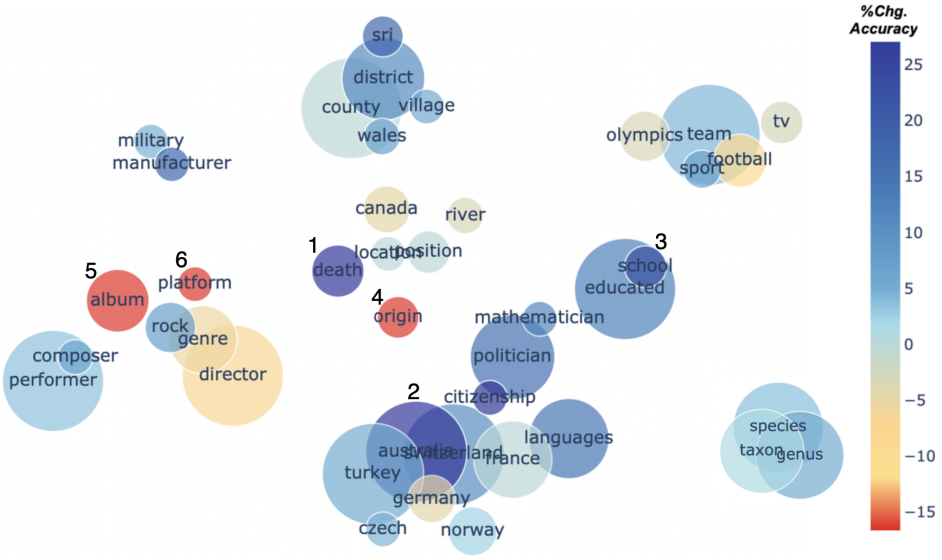

FIGURE 2 – Améliorations observées de la précision avec le modèle `Cooptiv PLM-A` sur le jeu de données T-REx par thème par rapport au modèle `Vanilla PLM-A`. La taille des classes est proportionnelle au nombre d'échantillons d'appartenance. Les 3 thèmes obtenant les meilleures performances ainsi que 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.