Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles
Avignon, France, 27 juin au 1er juillet 2022

Décontextualiser des plongements contextuels pour construire des thésaurus distributionnels

Olivier Ferret
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
olivier.ferret@cea.fr

RÉSUMÉ

Même si les modèles de langue contextuels sont aujourd'hui dominants en traitement automatique des langues, les représentations qu'ils construisent ne sont pas toujours adaptées à toutes les utilisations. Dans cet article, nous proposons une nouvelle méthode pour construire des plongements statiques à partir de modèles contextuels. Cette méthode combine la généralisation et l'agrégation des représentations contextuelles. Nous l'évaluons pour un large ensemble de noms en anglais dans la perspective de la construction de thésaurus distributionnels pour l'extraction de relations de similarité sémantique. Finalement, nous montrons que les représentations ainsi construites et les plongements statiques natifs peuvent être complémentaires.

ABSTRACT

Decontextualizing contextual embeddings for building distributional thesauri

While contextual language models are now dominant in the field of Natural Language Processing, the representations they build at the token level are not always suitable for all uses. In this article, we propose a new method for building word-level embeddings from contextual models. This method combines the generalization and the aggregation of token representations. We evaluate it for a large set of English nouns in the perspective of the building of distributional thesauri for extracting semantic similarity relations. Finally, we show that static embeddings and word-level embeddings can be complementary.

MOTS-CLÉS : Plongements statiques et contextuels, similarité, thésaurus distributionnel.

KEYWORDS: Static and contextual word embeddings, semantic similarity, distributional thesauri.

1 Introduction

L'introduction de modèles de langue contextuels tels qu'ELMo (Peters *et al.*, 2018) ou BERT (Devlin *et al.*, 2019) dans le domaine du traitement automatique des langues représente un changement majeur sur plusieurs plans. Du point de vue de la sémantique lexicale, l'un d'entre eux est le fait que ces modèles produisent des représentations au niveau des occurrences de mots, que nous appellerons ici tokens, et non des mots. Ce changement a globalement un impact positif sur les tâches de classification ou d'étiquetage de séquences mises en œuvre par des approches d'apprentissage automatique supervisé. Cependant, il soulève plus de difficultés pour les tâches typiquement traitées par des approches non supervisées et se concentrant sur le niveau du mot, comme l'extraction de relations sémantiques lexicales par exemple.

Volume 1 : conférence principale, pages 315–324.

Une façon de contourner ces difficultés est de construire des plongements au niveau du mot à partir d'un modèle de langue contextuel, ce qui a déjà été étudié par certains travaux. Dans le cadre de l'analyse des propriétés des modèles contextuels, Ethayarajh (2019) a ainsi proposé d'utiliser l'analyse en composantes principales (ACP) pour ce faire tandis que Bommasani *et al.* (2020) ont testé un ensemble plus large d'opérations. Le même type d'objectif se retrouve également dans (Vulić *et al.*, 2020b) et (Vulić *et al.*, 2020a), avec un accent mis sur l'étude des propriétés sémantiques. Enfin, Chronis & Erk (2020) ont exploré la question plus spécifique de plongements multi-prototypes destinés à rendre compte de la diversité des représentations des tokens. Par ailleurs, le problème que nous considérons est également lié à la construction de méta-plongements puisque le problème est de combiner plusieurs plongements dans les deux cas (Yin & Schütze, 2016; O'Neill & Bollegala, 2020).

Le travail de cet article [1] est plus particulièrement axé sur la production de plongements au niveau des mots à partir de modèles contextuels pour la construction de thésaurus distributionnels dans la perspective d'extraire des relations de similarité sémantique telles que les relations de synonymie. Plus précisément, nous présentons trois contributions principales : (i) nous proposons une nouvelle méthode pour produire des plongements de mots à partir de modèles contextuels en introduisant une forme de généralisation des représentations de tokens ; (ii) nous réalisons une évaluation à grande échelle de cette méthode dans un cadre complémentaire de ceux de Bommasani *et al.* (2020) ou Ethayarajh (2019) ; (iii) nous montrons qu'il est plus intéressant de considérer les plongements statiques natifs et les plongements statiques issus de modèles contextuels dans une perspective complémentaire que de remplacer les premiers par les seconds.

2 Méthode

Comme (Bommasani *et al.*, 2020) et (Ethayarajh, 2019), la méthode que nous proposons vise à agréger un ensemble de N_{tok} représentations de tokens pour chaque mot cible afin d'en produire une représentation. Chaque représentation de token correspond au plongement extrait pour une occurrence du mot cible dans une phrase à partir des résultats de l'encodage de cette phrase par un modèle de langue contextuel. Plus précisément, nous distinguons trois étapes dans la production d'une représentation au niveau du mot :

— la sélection des représentations de tokens considérées, destinée à éviter l'agrégation de représentations couvrant des sens trop hétérogènes du mot cible ;
— la généralisation des représentations de tokens sélectionnées, pour favoriser comme la première étape une certaine homogénéité des représentations agrégées ;
— enfin, la construction de la représentation du mot, s'opérant à l'instar de Bommasani *et al.* (2020) et Ethayarajh (2019) par l'agrégation des représentations de tokens.

2.1 Sélectionner les représentations des tokens

La première façon de restreindre la diversité des occurrences t_{ij} d'un mot w_i en termes de sens est de tirer ces occurrences d'un corpus homogène, ce que nous faisons dans les expériences de la section 3. Cependant, même si, comme le suggèrent McCarthy *et al.* (2004), la plupart des mots ont un sens

1. Une version étendue de ce travail est présentée dans (Ferret, 2022).

prédominant dans un corpus spécifique, leurs autres sens ne sont pas totalement négligeables. Par ailleurs, des travaux comme ceux de Garí Soler & Apidianaki (2021) ont montré que les modèles contextuels tels que BERT sont sensibles à la polysémie des mots au niveau des représentations qu'ils produisent. Pour contrôler ce facteur et tester son influence, nous faisons l'hypothèse que la moyenne des représentations $v_{(t_{ij})}$ des occurrences d'un mot devrait conduire à une représentation de ce mot, notée $moy(w_i)$, très proche de son sens prédominant. En considérant que nous choisissons un nombre fixe N_{sel_tok} d'occurrences pour chaque mot, nous proposons les options suivantes :

— *aléatoire* : dans cette option de base, les N_{sel_tok} tokens sont choisis de manière aléatoire parmi les N_{tok} tokens initialement sélectionnés pour le mot. C'est l'option généralement adoptée par les travaux existants dans ce domaine ;
— *prox_moy* : les tokens t_{ij} sont sélectionnés de telle sorte que la représentation $v_{t_{ij}}$ du token soit la plus proche de $moy(w_i)$, avec l'idée de favoriser l'homogénéité des tokens sélectionnés vers le sens prédominant du mot. C'est a priori la meilleure option en termes de précision ;
— *dist_moy* : à l'opposé de *prox_moy*, nous sélectionnons des tokens tels que $v_{t_{ij}}$ soit le plus éloigné de $moy(w_i)$ pour accroître la présence de sens mineurs du mot ;
— *uniforme* : l'idée est de tenir compte de la diversité des sens des mots en sélectionnant N_{sel_tok} tokens distribués uniformément en termes de similarité de $v_{t_{ij}}$ avec $moy(w_i)$. C'est a priori la meilleure option en termes de rappel.

2.2 Généraliser la représentation des tokens et construire celle des mots

Construire une représentation couvrant les tokens sélectionnés d'un mot nécessite, dans une certaine mesure, de gommer les différences de leurs représentations ou du moins, de renforcer leurs similitudes. Concrètement, il s'agit de rapprocher ces représentations les unes des autres tout en conservant l'essentiel de leurs spécificités. Cet objectif est proche du processus sous-jacent à plusieurs méthodes d'injection de relations de similarité sémantique dans les plongements statiques connues sous le terme générique de *retrofitting* (Faruqui *et al.*, 2015). Dans le cas des méthodes de *retrofitting*, le processus rapproche les représentations des mots impliqués dans des relations de similarité sémantique tandis que certaines méthodes éloignent en plus les représentations des mots faisant partie de relations de dissimilarité quand elles existent. Dans notre cas, nous appliquons une méthode de *retrofitting*, ici PARAGRAM (Wieting *et al.*, 2015), en considérant que les représentations des tokens d'un même mot sont implicitement liées par des relations de similarité, ce qui est effectivement vrai lorsque les tokens représentent différentes utilisations du même sens du mot et justifie notre première étape de sélection des tokens. Contrairement à Chronis & Erk (2020), nous ne regroupons pas les occurrences d'un mot dans des sens distincts et n'avons donc pas besoin d'introduire des relations de dissimilarité implicites entre les tokens appartenant à des sens différents.

Cette étape de généralisation et l'étape finale de construction de la représentation des mots peuvent être purement séquentielles ou bien partiellement jointes. Dans la première option, appelée *gen+agr*, une relation de pseudo-similarité est générée pour chaque paire de tokens d'un mot, une méthode de *retrofitting* est appliquée aux représentations de ces tokens en fonction de ces pseudo-relations et finalement, les représentations des tokens sont agrégées pour construire la représentation du mot.

Dans la seconde option, les représentations de tokens sont d'abord agrégées. Ensuite, l'étape de généralisation est appliquée à la fois aux représentations des tokens et au résultat de leur agrégation. L'intérêt de cette seconde option est d'inclure la représentation du mot dans le processus de généralisation et d'implémenter indirectement un nouveau type d'agrégation, de façon comparable à

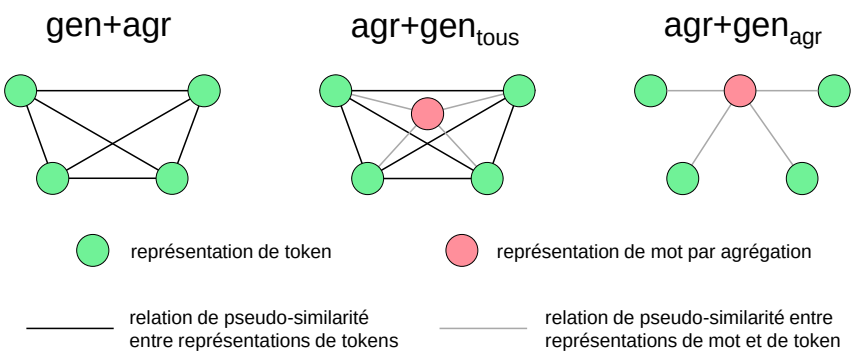

FIGURE 1 – Les trois stratégies de généralisation des représentations de tokens

(Ferret, 2018). Une première variante de cette seconde option, appelée *agr+gen*$_{tous}$, considère que le résultat de l'agrégation est une représentation supplémentaire du token et génère des relations de pseudo-similarité entre tous les tokens, y compris l'agrégat, comme dans la première option. La seconde variante, appelée *agr+gen*$_{agr}$, génère des pseudo-relations uniquement entre l'agrégat et tous les tokens, ce qui est une façon de concentrer l'opération de généralisation sur la représentation du mot.

Les trois stratégies résultant de ces différents choix sont illustrées par la figure 1, qui fait apparaître en noir les relations de pseudo-similarité entre représentations de tokens et en gris les relations de pseudo-similarité entre représentation de token et représentation de mot obtenue par agrégation des représentations de tokens. Il est à noter toutefois que ces deux types de relations ne sont pas différenciées du point de vue de l'algorithme de *retrofitting*. Par ailleurs, dans toutes ces stratégies, l'agrégation prend la forme d'une moyenne des représentations agrégées, en accord avec les conclusions de Bommasani *et al.* (2020).

3 Expérimentations

3.1 Cadre expérimental

Pour évaluer notre méthode, nous considérons deux modèles de langue contextuels pré-entraînés : BERT et CharacterBERT (El Boukkouri *et al.*, 2020), tous deux dans leur version `uncased` avec 12 couches (plus la couche d'entrée L0). Comme Bommasani *et al.* (2020), nous construisons la représentation de chaque token avec BERT en moyennant les représentations de ses sous-mots (*wordpieces*). L'intérêt de considérer CharacterBERT dans notre contexte est d'étudier l'impact de cette représentation des tokens puisque CharacterBERT peut directement produire une représentation pour chaque token. La construction de nos plongements au niveau du mot est fondée sur l'encodage par ces modèles d'un ensemble de phrases. Plus précisément, nous avons sélectionné aléatoirement un nombre maximal N_{tok} de 250 phrases pour chaque mot considéré w_i au sein d'AQUAINT, un corpus de 380 millions de mots composé d'articles de presse en anglais. Pour l'étape de sélection de la section 2.1, nous avons écarté les phrases de moins de 10 mots et de plus de 90 mots afin d'avoir un contexte significatif et ciblé lorsqu'au moins N_{sel_tok} ($N_{sel\_tok} = 10$) phrases remplissaient ces contraintes.

	$\mathbf{R}_{préc}$	MAP	P@1	P@2	P@5
CBERT-l4	15,6	18,0	22,0	15,9	9,2
CBERT-l4-tous	**16,1**	**18,4**	**22,5**	**16,3**	9,4
BERTrep-l4	12,2	14,0	17,2	12,5	7,3
BERTiso-l0	14,0	15,8	19,2	14,6	8,7
BERT-l5	15,6	17,9	21,8	16,0	**9,5**
fastText	15,5	**18,4**	21,9	15,7	9,2

TABLE 1 – Plongements statiques obtenus en moyennant des représentations de tokens

L'évaluation elle-même est effectuée dans le contexte de la construction de thésaurus distributionnels : pour chaque mot cible w_i, un ensemble de voisins distributionnels est sélectionné en calculant la similarité de w_i avec tous les autres mots cibles w_j par l'application de la mesure *cosinus* à leurs représentations et en classant ces mots selon la valeur décroissante de leur similarité avec w_i. Nous évaluons la pertinence de ce classement comme en recherche d'information avec la R-précision ($\mathrm{R}_{préc.}$), la MAP (Mean Average Precision) et les précisions à différents rangs (P@r). De manière similaire à des travaux tels que (Landauer & Dumais, 1997) ou (Freitag *et al.*, 2005) dont l'évaluation est fondée sur le paradigme TOEFL, notre référence est constituée de synonymes, provenant dans notre cas de WordNet (Miller, 1990) [2], avec 3 synonymes par mot en moyenne. L'évaluation est réalisée pour 10 305 noms couvrant un large spectre de fréquences [3].

3.2 Évaluation des références de base

La première étape de nos évaluations est l'application de l'approche de Bommasani *et al.* (2020), qui consiste à construire le plongement d'un mot en faisant la moyenne des plongements de ses occurrences dans un ensemble de phrases. Nous présentons les résultats de cette application, considérée comme une base de référence, dans le tableau 1 pour la meilleure couche de BERT (*BERT-l5*) et de CharacterBERT (*CBERT-l4*). Ces résultats sont obtenus à partir de 10 phrases choisies au hasard parmi les 250 phrases disponibles pour chacun de nos 10 305 noms.

Nous pouvons d'abord observer que les deux modèles obtiennent leurs meilleures performances pour pratiquement la même couche avec des valeurs très comparables pour toutes nos mesures d'évaluation. Cependant, il faut noter que CharacterBERT est entraîné dans les mêmes conditions que BERTrep, qui est équivalent en termes de modèle et de taille de corpus à BERT mais entraîné avec deux fois moins de lots. Les résultats de BERTrep dans le tableau 1, clairement inférieurs à ceux de BERT, suggèrent donc que l'entraînement de CharacterBERT pourrait être sous-optimal par rapport à celui de BERT, ce qui ne permet pas vraiment de conclure quant à l'intérêt de CharacterBERT par rapport à BERT pour la production de représentations de mots par moyennage. Le tableau 1 présente également les résultats de CharacterBERT sans la seconde étape de sélection des phrases (*CBERT-l4-tous*). Bien que la différence avec CBERT-l4 soit statistiquement significative [4], elle reste limitée. Nous ne considérerons donc par la suite que les performances avec la sélection de 10 phrases, qui constitue au

2. Plus précisément, pour chaque mot considéré, nous rassemblons tous les synonymes des synsets dont il fait partie.
3. Cet ensemble d'évaluation est repris de (Ferret, 2018).
4. La significativité des différences est jugée selon un test de Wilcoxon apparié avec p égal à $0,01$.

		$R_{préc}$	MAP	P@1	P@2	P@5
CBERT-moy (CBERT-l4)		15,6	18,0	22,0	15,9	9,2
CBERT-acp		15,6	17,9	22,0	15,9	9,2
CBERT-gen+agr		16,1	18,6	22,6	16,3	9,5
CBERT-agr+gen$_{tous}$		16,3	**18,9**	22,8	16,5	9,7
CBERT-agr+gen$_{agr}$		16,3	18,8	22,8	16,4	9,6
CBERT-retr-agr+gen$_{agr}$		15,6	18,0	22,0	15,9	9,2
BERT-moy (BERT-l5)		15,6	17,9	21,8	16,0	9,5
BERT-agr+gen$_{agr}$		16,2	18,8	22,5	16,1	10,1
fastText		15,5	18,4	21,9	15,7	9,2
CBERT-agr+gen$_{agr}$	uniforme	16,4	**18,9**	**22,9**	**16,7**	9,7
	dist_moy	**16,5**	**18,9**	**22,9**	16,6	**9,8**
	prox_moy	16,3	18,8	**22,9**	16,4	9,6

TABLE 2 – Évaluation de la méthode proposée pour construire des plongements statiques de mots

vu des résultats de *CBERT-l4-tous* et de *CBERT-l4* un compromis raisonnable entre coût et niveau de performance.

La ligne *BERTiso-l0* du tableau 1 rend compte quant à elle de l'approche « mot en isolation » de Vulić *et al.* (2020b), qui encode avec un modèle de langue contextuel une seule occurrence de chaque mot cible, considéré comme une phrase, et retient le plongement de cette occurrence comme plongement statique du mot. Le tableau 1 montre que les meilleurs résultats de cette approche, obtenus pour la couche L0 de BERT, sont significativement inférieurs à ceux de *BERT-l5*, l'approche plus répandue dite « en contexte », ce qui confirme les résultats de Vulić *et al.* (2020b) obtenus dans un cadre d'évaluation différent. Par conséquent, nous ne la considérerons pas dans ce qui suit. Enfin, la dernière ligne du tableau 1 (*fastText*) montre les résultats du modèle Skip-gram (Mikolov *et al.*, 2013) adopté par Vulić *et al.* (2020b) comme référence, entraîné sur Wikipédia à l'aide de fastText (Bojanowski *et al.*, 2017) [5]. Ce modèle obtient des résultats comparables à ceux des plongements construits à partir de CharacterBERT ou BERT, ce qui diffère des résultats de Bommasani *et al.* (2020) et Ethayarajh (2019), réalisés dans un contexte d'évaluation différent.

3.3 Évaluation de la méthode proposée

Dans le tableau 2, nous évaluons d'abord la méthode que nous proposons et ses différentes variantes avec la meilleure couche de CharacterBERT (CBERT-*) et nous testons finalement la meilleure variante sur la meilleure couche de BERT (*BERT-**) puisque les deux modèles sont proches dans la première évaluation. Notre référence de base est *CBERT-moy* pour CharacterBERT et *BERT-moy* pour BERT. Tous les résultats sont obtenus avec 10 phrases sélectionnées aléatoirement pour chaque mot. *CBERT-acp* correspond à l'application d'une ACP aux plongements de tokens proposée par Ethayarajh (2019) au lieu d'en faire la moyenne comme Bommasani *et al.* (2020). Le tableau 2 montre que les deux options sont équivalentes dans notre cas.

5. https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.en.zip

	$\mathbf{R}_{préc}$	MAP	P@1	P@2	P@5
CBERT-agr+gen$_{agr}$(uniforme)$_{haut}$	18,0	20,5	26,6	19,8	12,0
CBERT-agr+gen$_{agr}$(uniforme)$_{bas}$	14,8	17,3	19,3	13,4	7,3
fastText$_{haut}$	14,5	16,8	22,0	15,9	9,7
fastText$_{bas}$	16,4	20,0	21,8	15,4	8,6
CombSum	18,6	21,5	25,9	19,0	11,1

TABLE 3 – Comparaison des plongements construits à partir de CharacterBERT (cf. *uniforme* dans le tableau 2) et des plongements Skip-gram (*fastText*) en fonction de la fréquence des mots (*haut* ou *bas*)

En ce qui concerne notre méthode, nous observons tout d'abord que nos trois variantes surpassent significativement notre référence. Cette amélioration n'est pas importante mais elle est suffisante pour dépasser aussi les plongements Skip-gram de manière significative. Nous observons également que nos trois variantes sont très proches mais que séparer la généralisation et l'agrégation (*CBERT-gen+agr*) est une option légèrement moins bonne que de les joindre. Les deux variantes jointes sont assez équivalentes en termes d'évaluation mais *CBERT-agr+gen*$_{agr}$ est moins gourmande en calcul en raison d'un nombre beaucoup plus faible de relations de pseudo-similarité. Les performances de *CBERT-retr-agr+gen*$_{agr}$, qui remplace PARAGRAM par la méthode de Faruqui *et al.* (2015), confirment l'intérêt de PARAGRAM pour la tâche de généralisation. Enfin, *BERT-agr+gen*$_{agr}$ montre que les résultats obtenus pour CharacterBERT peuvent être globalement transposés à BERT.

Le dernier aspect de la méthode proposée à évaluer est la stratégie de sélection des tokens utilisés pour construire les représentations des mots. Cette évaluation est présentée dans les trois dernières lignes du tableau 2 pour la variante *CBERT-agr+gen*$_{agr}$, toutes les autres lignes BERT-* ou CBERT-* correspondant à une sélection *aléatoire*. Contrairement à ce que l'on peut attendre du caractère fortement contextualisé de la représentation des tokens, leur sélection en fonction de leur proximité avec le sens dominant du corpus n'a pas une forte influence sur les résultats. Un léger avantage est observé pour les stratégies favorisant la diversité entre les tokens (*uniforme* et *dist_moy*) mais les résultats sont probablement limités par le fait que, comme démontré par Ethayarajh (2019), la contextualisation est plus forte pour les couches élevées, qui ne sont pas les meilleures ici.

3.4 Analyses complémentaires

Au-delà des résultats globaux de la section précédente, le tableau 3 offre une autre vision du rapport entre les plongements statiques de fastText et ceux construits à partir de CharacterBERT en déclinant les différentes mesures selon la médiane fréquentielle du vocabulaire. Il en ressort ainsi un constat important : alors que les plongements issus de CharacterBERT dépassent incontestablement les plongements fastText pour les mots de haute fréquence, la tendance est opposée pour les mots de basse fréquence, illustrant qu'au-delà de leurs performances respectives, les plongements statiques natifs et les plongements issus de modèles contextuels présentent des propriétés complémentaires concernant la fréquence des mots. Il s'agit typiquement d'une configuration intéressante pour l'application de méthodes d'ensemble.

La dernière ligne du tableau 3 confirme cet intérêt en présentant les résultats d'une telle application selon une approche de fusion tardive. Plus précisément, cette approche, fondée sur (Curran & Moens, 2002), consiste à fusionner les thésaurus construits à partir des deux types de plongements en

fusionnant les listes de voisins distributionnels associées à chacune de leurs entrées. Cette fusion est effectuée par la méthode CombSum (Fox & Shaw, 1994) fondée sur des valeurs de similarité normalisées avec la méthode Zero-one (Wu *et al.*, 2006; Lee, 1997). Comme le montre le tableau 3, le thésaurus résultant surpasse nettement les deux thésaurus initiaux, ce qui confirme la complémentarité des deux plongements dont ils sont issus et souligne le fait qu'ils sont plus complémentaires que concurrents.

4 Conclusion et perspectives

Dans cet article, nous avons proposé une nouvelle méthode pour construire des plongements statiques à partir de plongements contextuels. Cette méthode se fonde sur un triple processus commençant par la sélection de représentations de tokens produites par un modèle de langage contextuel, leur généralisation, et enfin, leur agrégation pour construire des représentations au niveau du mot. Les résultats que nous avons présentés montrent que la méthode que nous proposons dépasse les méthodes de référence de Bommasani *et al.* (2020), Ethayarajh (2019) et Vulić *et al.* (2020b) dans une évaluation de type TOEFL et peut également être comparée favorablement à une méthode produisant directement des plongements statiques telle que fastText. Par ailleurs, une analyse de ces résultats du point de vue de la fréquence des mots montrent que les plongements statiques natifs et ceux construits à partir de modèles de langage contextuels sont davantage complémentaires que concurrents et que leur association peut s'avérer fructueuse pour une tâche telle que l'extraction de synonymes.

Dans ce travail, nous nous sommes concentrés sur la construction de représentations de mots à partir de représentations de tokens mais sans envisager la possibilité d'associer les représentations de tokens issues de différentes couches du modèle de langage contextuel considéré. Le travail de Vulić *et al.* (2020b) suggère que l'association de représentations provenant de différentes couches d'un tel modèle peut être intéressante. Nous envisageons d'étudier cette possibilité en allant au-delà de l'utilisation de l'approche de base consistant à faire la moyenne des représentations, en appliquant soit le type de processus que nous avons proposé dans cet article, soit des méthodes fondées sur la projection de différents espaces de représentation dans un espace partagé, comme (Caciularu *et al.*, 2021).

Remerciements

Le travail présenté dans cet article a été réalisé dans le cadre du projet ADDICTE [6] (Analyse distributionnelle en domaine spécialisé), financé par l'Agence Nationale de la Recherche (ANR-17-CE23-0001). Nous remercions par ailleurs Hicham El-Boukkouri pour son aide précieuse concernant le modèle CharacterBERT.

Références

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

6. https://anr-addicte.ls2n.fr/

DOI : 10.1162/tacl_a_00051.

BOMMASANI R., DAVIS K. & CARDIE C. (2020). Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4758–4781, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.431.

CACIULARU A., DAGAN I. & GOLDBERGER J. (2021). Denoising word embeddings by averaging in a shared space. In **SEM 2021 : The Tenth Joint Conference on Lexical and Computational Semantics*, p. 294–301, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.starsem-1.28.

CHRONIS G. & ERK K. (2020). When is a bishop not like a rook ? When it's like a rabbi ! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, p. 227–244, Online : Association for Computational Linguistics.

CURRAN J. R. & MOENS M. (2002). Improvements in automatic thesaurus extraction. In *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, p. 59–66, Philadelphia, USA.

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

EL BOUKKOURI H., FERRET O., LAVERGNE T., NOJI H., ZWEIGENBAUM P. & TSUJII J. (2020). CharacterBERT : Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *28th International Conference on Computational Linguistics (COLING 2020)*, p. 6903–6915, Barcelona, Spain (Online : International Committee on Computational Linguistics.

ETHAYARAJH K. (2019). How Contextual are Contextualized Word Representations ? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, p. 55–65, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1006.

FARUQUI M., DODGE J., JAUHAR S. K., DYER C., HOVY E. & SMITH N. A. (2015). Retrofitting Word Vectors to Semantic Lexicons. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT 2015)*, p. 1606–1615, Denver, Colorado.

FERRET O. (2018). Using pseudo-senses for improving the extraction of synonyms from word embeddings. In *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, p. 351–357, Melbourne, Australia : Association for Computational Linguistics.

FERRET O. (2022). Building static embeddings from contextual ones : Is it useful for building distributional thesauri ? In *13th Language Resources and Evaluation Conference (LREC 2022)*, Marseille, France.

FOX E. A. & SHAW J. A. (1994). Combination of multiple searches. In *2nd Text REtrieval Conference (TREC-2)*, volume 243 : NIST.

FREITAG D., BLUME M., BYRNES J., CHOW E., KAPADIA S., ROHWER R. & WANG Z. (2005). New experiments in distributional representations of synonymy. In *Ninth Conference on Computational Natural Language Learning (CoNLL 2005)*, p. 25–32, Ann Arbor, Michigan, USA.

GARÍ SOLER A. & APIDIANAKI M. (2021). Let's Play Mono-Poly : BERT Can Reveal Words' Polysemy Level and Partitionability into Senses. *Transactions of the Association for Computational Linguistics*, **9**, 825–844. DOI : 10.1162/tacl_a_00400.

LANDAUER T. K. & DUMAIS S. T. (1997). A solution to Plato's problem : the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, **104**(2), 211–240.

LEE J. H. (1997). Analyses of multiple evidence combination. In *20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, p. 267—-276, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/258525.258587.

MCCARTHY D., KOELING R., WEEDS J. & CARROLL J. (2004). Finding Predominant Word Senses in Untagged Text. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, p. 279–286, Barcelona, Spain. DOI : 10.3115/1218955.1218991.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

MILLER G. A. (1990). WordNet : An On-Line Lexical Database. *International Journal of Lexicography*, **3**(4).

O'NEILL J. & BOLLEGALA D. (2020). Meta-embedding as auxiliary task regularization. In *ECAI*, p. 2124–2131 : IOS Press.

PETERS M., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2018)*, p. 2227–2237, New Orleans, Louisiana, USA : Association for Computational Linguistics.

VULIĆ I., BAKER S., PONTI E. M., PETTI U., LEVIANT I., WING K., MAJEWSKA O., BAR E., MALONE M., POIBEAU T., REICHART R. & KORHONEN A. (2020a). Multi-SimLex : A Large-Scale Evaluation of Multilingual and Crosslingual Lexical Semantic Similarity. *Computational Linguistics*, **46**(4), 847–897. DOI : 10.1162/coli_a_00391.

VULIĆ I., PONTI E. M., LITSCHKO R., GLAVAŠ G. & KORHONEN A. (2020b). Probing Pretrained Language Models for Lexical Semantics. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, p. 7222–7240, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.586.

WIETING J., BANSAL M., GIMPEL K. & LIVESCU K. (2015). From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics*, **3**, 345–358.

WU S., CRESTANI F. & BI Y. (2006). Evaluating score normalization methods in data fusion. In *Third Asia Conference on Information Retrieval Technology (AIRS'06)*, p. 642–648 : Springer-Verlag.

YIN W. & SCHÜTZE H. (2016). Learning word meta-embeddings. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, p. 1351–1360.