Génération automatique de définitions pour le français

Timothee Mickus, Mathieu Constant, Denis Paperno

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Génération automatique de définitions pour le français

Timothee Mickus[1] Mathieu Constant[1] Denis Paperno[2]
(1) ATILF, 44 Avenue de la Libération, 54000 Nancy, France
(2) Universiteit Utrecht, Domplein 29, 3512 JE Utrecht, Netherlands
tmickus@atilf.fr, mconstant@atilf.fr, d.paperno@uu.nl

RÉSUMÉ

La génération de définitions est une tâche récente qui vise à produire des définitions lexicographiques à partir de plongements lexicaux. Nous remarquons deux lacunes : (i) l'état de l'art actuel ne s'est penché que sur l'anglais et le chinois, et (ii) l'utilisation escomptée en tant que méthode d'évaluation des plongements lexicaux doit encore être vérifiée. Pour y remédier, nous proposons un jeu de données pour la génération de définitions en français, ainsi qu'une évaluation des performances d'un modèle de génération de définitions simple selon les plongements lexicaux fournis en entrée.

ABSTRACT

Definition Modeling in French

Definition modeling is a recent task that aims at producing dictionary definitions based on word embeddings. We observe two gaps : (i) the current state of the art has yet to tackle languages other than English or Chinese and (ii) the purported usability as an evaluation method for word embeddings has yet to be verified. Hence we propose a dataset for French definition modeling and evaluate how using different input embeddings impacts the performances of a simple definition modeling system.

MOTS-CLÉS : Génération de définitions – plongements lexicaux – sémantique distributionnelle.

KEYWORDS: Definition modeling – word embeddings – distributional semantics.

1 Introduction

La *génération de définitions* (Noraset *et al.*, 2017, ou '*definition modeling*') vise à convertir un jeu de plongements lexicaux en un jeu de définitions équivalentes, telles qu'elles pourraient apparaître dans un dictionnaire. Fournir à un modèle de génération de définitions le vecteur du mot "*répétitivité*" devrait produire une sortie telle que "*Qualité, caractère de ce qui est répétitif (événements, gestes, etc.).*". [1] Suivant les conventions lexicographiques, on appellera ici le mot à définir le *definiendum* (pluriel : *definienda*).

Un ensemble conséquent de travaux a établi que les plongements lexicaux correspondent à des représentations de sémantique distributionnelle (Lenci, 2018). Ces dernières sont par conséquent explicitement mises en équivalence, dans les modèles de génération de définitions, avec les descriptions traditionnelles du sens. Le linguiste peut donc voir dans la génération de définitions une expérience montrant que la représentation distributionnelle d'un mot est conforme aux intuitions

1. Définition du wiktionnaire (fr.wiktionary.org)

Volume 2 : Traitement Automatique des Langues Naturelles, pages 66–80.
HAL : hal-02784756.

des locuteurs. [2] Pour l'informaticien, cette tâche permet de vérifier qu'un plongement lexical a été correctement pré-entraîné : si un vecteur de mot est une représentation correcte d'un mot, on s'attend à ce qu'il contienne toute l'information nécessaire afin de reconstruire une description du sens plus traditionnelle, en particulier la définition du mot telle qu'elle apparaît dans un dictionnaire.

Cet usage escompté de la génération de définitions comme outil d'évaluation faisait partie intégrante de la proposition initiale de Noraset *et al.* (2017) ; or aucune vérification n'en a été réalisée. La littérature subséquente sur le sujet a considéré comme acquise cette utilisation : aucun des travaux portés à notre connaissance ne compare deux jeux de plongements lexicaux, ni n'étudie l'importance des plongements pour la production de définitions. Pour y remédier, nous proposons de comparer les performances d'une architecture selon les représentations vectorielles fournies en entrée.

L'emploi suggéré par Noraset *et al.* (2017) n'est pas le seul attrait que présente cette tâche. On peut évoquer plusieurs applications pratiques à la génération de définitions : par exemple produire des ébauches de dictionnaires pour des langues peu dotées ou bien servir d'aide à la lecture pour les apprenants d'une langue étrangère. Ces usages, cependant, requièrent d'étudier les capacités des modèles sur plusieurs langues. Toute la littérature concernant la génération de définitions s'est pourtant focalisée sur l'anglais, à l'exception notoire de Yang *et al.* (2019) qui étudient le chinois. Pour répondre en partie à cette seconde lacune, nous produisons un jeu de données pour le français.

Le présent article est structuré comme il suit : nous mentionnerons d'abord en section 2 l'état de l'art dans le domaine de la géneration de définitions, avant de détailler le jeu de données que nous proposons pour le français (section 3), les modèles étudiés (section 4), les résultats préliminaires dont nous disposons (section 5) et enfin les erreurs typiquement commises par nos modèles (section 6). Nous terminerons par quelques perspectives en section 7.

2 État de l'art

L'utilisation de dictionnaires est une idée féconde et ancienne en traitement automatique des langues. Dans le cadre de cet exposé, on distinguera trois types de travaux. Une première catégorie contiendrait les travaux qui cherchent à modéliser ces ressources lexicales, que ce soit sous forme d'ontologie (Chodorow *et al.*, 1985), de graphes (Gaume *et al.*, 2014) ou autres. Une seconde s'intéresse davantage à comment exploiter les définitions qu'ils contiennent : par exemple, Gaume *et al.* (2004), qui indiquent qu'un dictionnaire permet de désambiguïser un mot ou encore Hill *et al.* (2016b), qui se penchent notamment sur la tâche de dictionnaire inversé, correspondant à trouver un mot à partir d'une définition ; on pensera aussi à Hill *et al.* (2016a) qui cherchent à produire un modèle de sémantique compositionnelle à partir de dictionnaires. Une troisième catégorie cherche à étoffer ces ressources, parmi lesquels on peut citer Sierra *et al.* (2015), qui proposent une méthodologie pour extraire automatiquement des définitions dans du texte 'brut'. De nombreuses entreprises de recherches s'inscrivent dans plusieurs de ces registres à la fois : par exemple, Bosc & Vincent (2018) proposent un système d'auto-encodage des définitions qui peut être vu à la fois comme une modélisation du dictionnaire, ainsi qu'une manière d'utiliser ceux-ci afin de produire des plongements lexicaux. Le programme de recherche qu'introduit la tâche de génération de définitions peut être considéré comme

2. Nous notons que l'établissement de dictionnaires (ou documents similaires) est attesté à travers les civilisations et les époques : à titre d'exemple, on peut citer le Er ya de la Chine antique, le glossaire de Cormac, texte irlandais du X^{ème} siècle, ou encore le projet moderne Wiktionary (wiktionary.org). Contrairement à d'autres ressources "expertes" telles que WordNet (Fellbaum, 1998), ces différentes entreprises ne sont pas guidées par les intuitions des linguistes spécialistes du domaine.

relevant de ces trois catégories de travaux : il propose de développer un modèle neuronal convertissant des *definienda* en définitions, ce qui servirait à la fois à évaluer des représentations vectorielles de mots et à compléter des dictionnaires existants.

La génération de définitions a été introduite par Noraset *et al.* (2017), qui la conçoivent avant tout comme une tâche d'évaluation extrinsèque de jeux de plongements lexicaux ; en cela, elle est à rapprocher de la vaste littérature dédiée à l'analyse des jeux de plongements lexicaux (Levy & Goldberg, 2014a,b; Arora *et al.*, 2018; Batchkarov *et al.*, 2016; Swinger *et al.*, 2018, entre autres), qui plus traditionnellement se concentre sur la structure de l'espace vectoriel et de l'application des vecteurs à la tâche d'analogie formelle (Mikolov *et al.*, 2013b; Gladkova *et al.*, 2016; Grave *et al.*, 2018).

La formulation initiale de la tâche par Noraset *et al.* (2017) considère le *definiendum* isolé de son contexte : Gadetsky *et al.* (2018) remarquent que ceci est problématique pour les mots ambigus ou polysémiques, pour lesquels établir le sens requiert d'avoir accès au contexte ; ils utilisent par conséquent des plongements contextualisés AdaGram (Bartunov *et al.*, 2016). Mickus *et al.* (2019) soulignent qu'une architecture de type "encodeur-décodeur" permet de traiter de manière uniforme les *embeddings* contextualisés et non-contextualisés, ainsi que les unités polylexicales et les mots simples. Zhu *et al.* (2019) et Chang *et al.* (2018) proposent de décomposer les sens d'un mot en représentations vectorielles creuses (Arora *et al.*, 2018). Chang *et al.* (2018) critiquent de plus que ces modèles sont relativement difficiles à interpréter, ce qui nuit à l'utilité de la tâche en tant qu'outil d'évaluation (d'où leur emploi de représentations creuses). Zhang *et al.* (2019) explorent différentes architectures qui permettent de générer non seulement la définition, mais aussi un exemple d'utilisation du *definiendum*.

Tous les travaux précédents étudient spécifiquement le cas de l'anglais. Yang *et al.* (2019) proposent un jeu de données pour le chinois, qui inclut des caractères sémantiquement liés au *definiendum* manuellement annotés. Tout comme le jeu proposé par Noraset *et al.* (2017), le jeu de Yang *et al.* (2019) ne contient pas d'exemples d'usage des *definienda*.

3 Jeu de données

Le jeu de données que nous proposons est tiré de GLAWI (Hathout & Sajous, 2016), une version du wiktionnaire français au format XML. GLAWI recense pour chaque mot-forme les parties du discours applicables, et pour chacune de celles-ci les sens possibles, accompagnés d'une définition (balise `<gloss>`) ainsi qu'éventuellement d'un exemple d'usage (balise `<example>`). Nous retirons les exemples d'usage où le *definiendum* n'est pas présent dans le contexte. [3] Nous convertissons le jeu de données entier en caractères minuscules. L'emploi du wiktionnaire comme source de données implique que notre jeu contient aussi des expressions polylexicales. [4] L'utilisation de *definienda* polylexicaux n'est pas standard en génération de définitions : comme le but premier de cette tâche est l'évaluation extrinsèque de plongements lexicaux, la capacité à prendre en compte une entrée séquentielle est souvent sacrifiée en faveur d'une architecture liant directement une représentation vectorielle à une production. [5]

3. Comme Mickus *et al.* (2019), nous conservons les exemples d'usages où une variante fléchie du *definiendum* est présente ; pour ce faire nous utilisons la librairie python `spacy` (https://spacy.io/).

4. Seuls les *definienda* polylexicaux réalisés de manière continue ont étés conservés.

5. Laissant un instant de côté la formalisation initiale de la tâche en tant qu'outil d'évaluation, il est notoire que l'avènement de plongements contextualisés est allé de pair avec l'accroissement de la demande en ressources computationnelles pour l'entraînement. Un nombre important de ces modèles, pour y répondre, font usage de tokenisation en sous-mots afin de limiter

# d'exemples	*Definienda* distincts	Taille moy. des exemples	Taille moy. des définitions
232037	100288	25,36	12,01

TABLE 1: Distribution des définitions utilisables de GLAWI.

La table 1 présente quelques statistiques concernant notre jeu de données. L'exemple d'usage est généralement deux fois plus long que la définition à produire. De plus, 12% des exemples correspondent à un *definiendum* polylexical ; la longueur moyenne des exemples d'utilisations et des définitions est relativement similaire dans les deux cas des *definienda* polylexicaux et monolexicaux. À titre de comparaison, les jeux de données proposés par Yang *et al.* (2019) contiennent un peu plus de 100000 exemples pour le chinois. Quant à l'anglais, le jeu de données de Gadetsky *et al.* (2018) contient un peu plus de 122000 exemples (dont 20% de cas où le *definiendum* ne peut pas être extrait de l'exemple d'utilisation associé), Zhu *et al.* (2019) proposent un jeu de données d'un peu plus de 120000 exemples et enfin Zhang *et al.* (2019) proposent près de 300000 exemples.

Nous séparons le jeu de données en trois sous-ensembles, un d'entraînement (80 %), un de validation (10 %) et un de test (10 %) ; nous utilisons la même partition dans toutes les expériences mentionnées plus bas. La partition est réalisée de manière à ce que les *definienda* soient uniques au sous-ensemble où ils apparaissent. Les sous-ensembles totalisent respectivement 185363, 23178 et 23496 exemples.

4 Modèle et jeux de plongements lexicaux

4.1 Architecture de génération de définitions

Nous reprenons le modèle de Mickus *et al.* (2019), qui permet de traiter uniformément les unités polylexicales et monolexicales. Ce modèle est entraîné à générer des définitions à partir des exemples d'usages associés aux définitions ; nous y renvoyons le lecteur pour tout détail complémentaire. À un niveau formel, il correspond à une architecture Transformer (Vaswani *et al.*, 2017), dotée de vecteurs "marqueurs", ici $\vec{D}$ et $\vec{C}$ distinguant respectivement le *definiendum* de son contexte. Dans le cadre du présent travail, nous ajoutons systématiquement le *definiendum* au début de l'exemple d'usage, précédé d'un token spécial `[DDUM]`, et suivi d'un autre token spécial `[DENS]`.[6]

Une illustration de l'architecture encodeur-décodeur utilisée par ce modèle est présentée dans la Figure 1 : la séquence composée du token spécial `[DDUM]`, suivi du *definiendum* suivi du token spécial `[DENS]`, suivi de l'exemple d'usage est passée à l'encodeur (en rouge dans la figure), qui la convertit en une séquence de représentations intermédiaires $\vec{r_1}, \ldots, \vec{r_n}$, désignées collectivement comme "banque d'attention" (en vert dans la figure). Le décodeur (en jaune dans la figure) est prompté

la taille du vocabulaire. Pour prendre un exemple récent, CamemBERT (Martin *et al.*, 2019) et FlauBERT (Le *et al.*, 2019), qui pré-entraînent tous deux l'architecture BERT (Devlin *et al.*, 2019) pour le français, font respectivement usage des algorithmes SentencePiece (Kudo & Richardson, 2018) et BPE (Sennrich *et al.*, 2016). Or l'éclatement d'un mot en sous-mots conduit à remplacer une entrée composée d'une seule représentation vectorielle par une série ordonnée de vecteurs : c'est-à-dire conduit à considérer certaines unités monolexicales comme polylexicales.

6. L'architecture Transformer faisant un usage systématique de connections résiduelles entre chacune des couches de l'encodeur, interrompues seulement par des opérations de normalisation, on peut garantir que la représentation intermédiaire $\vec{r_k}$ correspondant au k-ième token de la séquence gardera trace de la composition linéaire du vecteur fourni en entrée. Par conséquent les représentations intermédiaires correspondant aux éléments de contexte résideront dans un sous-espace vectoriel distinct de celui correspondant aux représentations de *definienda*.

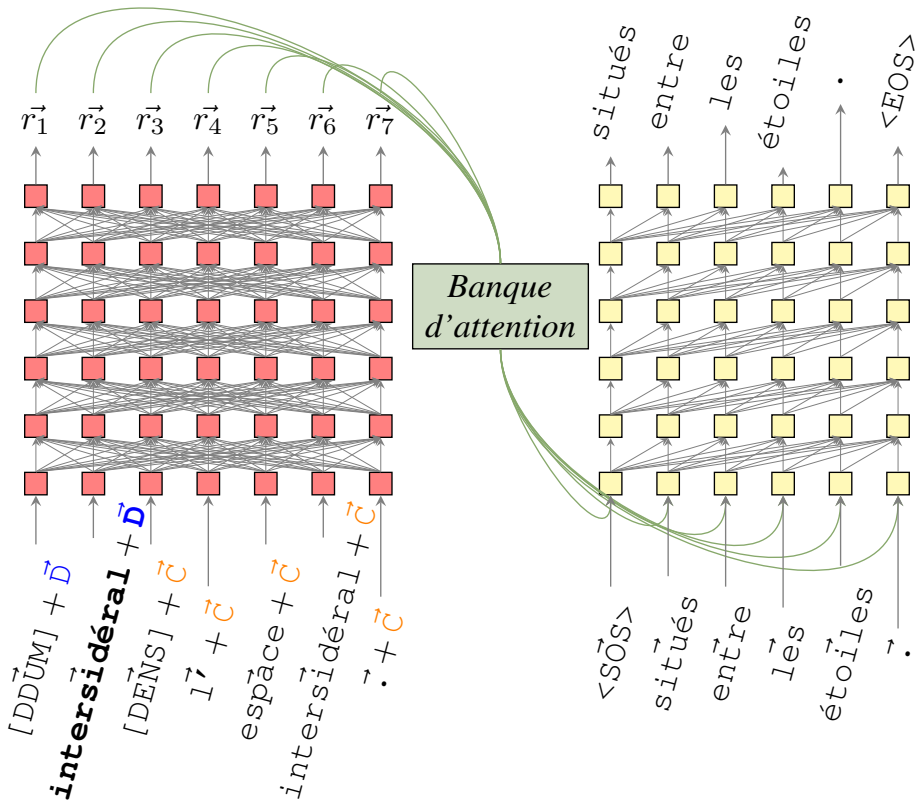

FIGURE 1: Vue d'ensemble de l'architecture du modèle de génération de définitions.

avec un symbole spécial indiquant le début de séquence <SOS> ; la génération se termine lorsqu'il produit un symbole de fin de séquence <EOS>. Lors de l'apprentissage, la définition tirée du jeu de données est fournie en entrée du décodeur ; pendant la génération, l'entrée à l'étape t correspond au symbole généré à l'étape précédente $t-1$. Le décodeur a toujours accès à la banque d'attention, qu'il utilise à travers des mécanismes d'attention multi-tête. Comme les modèles de génération de définitions sont censés servir à l'évaluation des plongements lexicaux, nous gelons les poids des représentations vectorielles.

Nous utilisons 12 couches dans l'encodeur et le décodeur, un *warmup* de 10000 étapes, un taux d'apprentissage de 1 et un *label smoothing* de 0,15 ; autrement les hyperparamètres correspondent à ceux initialement suggérés dans Mickus *et al.* (2019). Ces paramètres correspondent à la configuration optimale trouvée lors d'expériences préliminaires.

4.2 Plongements lexicaux

Nous comparons plusieurs architectures de plongements lexicaux : word2vec (Mikolov *et al.*, 2013a, CBOW), GloVe (Pennington *et al.*, 2014) et FastText (Bojanowski *et al.*, 2017) ; toutes nos représentations sont de dimension 300. Pour ce qui est de word2vec et GloVe, nous utilisons la concaténation de FRCOW (Schäfer, 2015) et d'un dump de Wikipedia français parsé par Coavoux (2017) [7] comme corpus d'entraînement. Ces deux corpus sont mis en minuscules, pour un total de 7,25 milliards de tokens. Pour word2vec, nous utilisons 20 contre-exemples et une fenêtre de 10 mots et parcourons

7. Disponible à l'adresse suivante : http://www.llf.cnrs.fr/wikiparse/

aléatoires	word2vec	GloVe	FastText MC	FastText FB
0,00	36,46	58,32	57,38	68,63

TABLE 2: Analogie formelle : précision des différentes représentations vectorielles (pourcentages).

le corpus sur 10 itérations ; les vecteurs GloVe sont entraînés pendant 10 itérations avec les hyperparamètres proposés dans le script d'exemple fourni par Pennington *et al.* (2014). Pour FastText, nous utilisons deux jeux de plongements : un jeu entraîné sur ce même corpus pendant 5 itérations ("FastText MC"), et celui fourni par Grave *et al.* (2018) ("FastText FB"). Ce jeu de plongements FastText FB est décrit en davantage de détails par Grave *et al.* (2018) ; nous soulignerons seulement ici que le corpus d'apprentissage utilisé contient un dump de Wikipedia (1,11 milliard de tokens) ainsi qu'un sous ensemble de Common Crawl (68,36 milliards de tokens), c'est-à-dire un ensemble de données d'apprentissage environ 10 fois plus grand. Comparer ces deux jeux devrait nous permettre d'appréhender la sensibilité de la génération de définitions aux choix d'hyperparamètres et à l'accès aux données. Enfin, nous évaluons aussi une matrice initialisée aléatoirement $M_{ij} \sim \mathcal{N}(0, 1)$ avec la même dimension que nos plongements lexicaux.

La précision atteinte par ces différentes représentations sur le jeu d'analogie formelle de Grave *et al.* (2018) est indiquée dans la table 2. [8] Évaluer les performances des représentations vectorielles à l'aide d'une méthode répandue comme l'analogie formelle nous permet d'étudier si la génération de définitions peut servir en tant que méthode d'évaluation des plongements lexicaux. Ceci nous permet d'estimer *a priori* un degré raisonnable de variation des performances des différents jeux de plongements : si les résultats obtenus sur l'analogie formelle et ceux obtenus en génération de définitions divergent radicalement, il nous faudra soit remettre en cause la validité de la tâche générative en tant que méthode d'évaluation, soit avancer que les aspects mesurés par ces deux méthodes diffèrent radicalement. Un tel scénario est effectivement envisageable : rien ne garantit que la régularité linéaire de l'espace sémantique des plongements soit utile à la génération de définitions. Soulignons toutefois que les deux tâches devraient être toutes deux sensibles à la qualité des représentations vectorielles testées ; aussi est-il raisonnable d'escompter une certaine congruence de leurs résultats respectifs.

5 Résultats

Nous évaluons nos résultats à l'aide de deux métriques : perplexité et score BLEU (Papineni *et al.*, 2002). Ces métriques sont couramment utilisées en génération automatique. La perplexité est censée dépeindre l'incertitude du modèle de pouvoir engendrer la cible. Elle se calcule comme l'exponentiation de l'entropie croisée : par conséquent on préférera les modèles qui minimisent la perplexité. Les scores BLEU calculent la similarité du vocabulaire employé dans la cible et dans la production du modèle : les modèles où le score BLEU est maximal sont *a priori* à privilégier.

Les performances des modèles que nous étudions sont rapportées dans la table 3. Pour chaque jeu de représentations vectorielles, nous rapportons les scores de perplexité et les scores BLEU sur l'ensemble de test. Deux remarques ressortent immédiatement de l'étude des scores de perplexité. Premièrement,

8. Nous supprimons les doublons du jeu de Grave *et al.* (2018), ainsi que les exemples contenant la paire "*son*" — "*sa*", qui correspond à un contraste de genre grammatical et non de genre social. 646 exemples sont concernés. Nous mettons le jeu entier en minuscules.

Plongements	Perplexité	BLEU
aléatoires	83,70	19,90
word2vec	52,13	30,60
GloVe	48,55	29,00
FastText MC	45,04	30,50
FastText FB	47,84	32,80

TABLE 3: Génération de définitions : résultats généraux (ensemble de test).

le modèle entraîné sur des représentations aléatoires produit des résultats bien moins bons que les modèles entraînés sur des plongements lexicaux. Ceci suggère que la génération de définitions répond aux critères minimum pour être utilisée afin d'évaluer les plongements lexicaux. Deuxièmement, les différences de performance entre jeux de plongements lexicaux sur la tâche d'analogie formelle (cf. table 2) ne sont pas celles observées pour la génération de définitions : si l'ordre des trois architectures varie peu, la différence entre les vecteurs GloVe et FastText est moindre que celle observée pour l'analogie formelle ; [9] de plus, le modèle FastText que nous avons entraîné s'avère comparable à celui distribué par Grave *et al.* (2018), alors qu'on observait une différence claire en analogie formelle. Les scores BLEU donnent un élément de contraste à ce que nous indique la perplexité. Ici, les représentations aléatoires sont toujours nettement inférieures ; cependant les vecteurs GloVe sont cette fois-ci perçus comme moins bons que les vecteurs word2vec. Nous remarquons de plus un compromis entre le score BLEU et la perplexité des deux modèles FastText, ce qui n'aide pas à départager les deux métriques.

Notons toutefois deux points importants. D'une part, les hyperparamètres de nos modèles n'ont pas étés spécifiquement réglés : une recherche plus extensive pourrait avoir un impact significatif sur ces résultats. D'autre part, une autre interprétation de ces résultats serait que les modèles disposent de suffisamment de ressources pour gommer les différences entre chaque jeu de plongements, ce qui gênerait l'utilisation de la génération de définitions en tant que tâche d'évaluation.

Afin de compléter le tableau que dressent les deux métriques automatiques, nous pouvons comparer également les scores BLEU obtenus en récupérant simplement la définition correspondant à l'entrée la plus proche dans l'ensemble d'entraînement. Ceci nous permet d'établir un point de comparaison utile, en indiquant le score qui serait obtenu par un modèle qui copierait simplement des exemples déjà rencontrés. Nous étudions deux variantes pour cette ligne de référence. La première consiste à récupérer l'entrée globalement la plus similaire, ce que l'on peut mesurer approximativement en calculant la similarité entre exemples d'utilisation fournis lors de l'inférence et exemples d'utilisation vus lors d'apprentissage ; nous mesurons la similarité des exemples d'utilisation à l'aide du cosinus des vecteurs moyens des mots qu'ils contiennent. La seconde repose sur l'idée que des mots aux sens similaires auront des définitions semblables ; nous comparons donc aussi les définitions dont les *definienda* sont les plus similaires. Nous calculons ces deux lignes de référence à partir de l'ensemble de validation.

Les résultats correspondant à ces deux lignes sont décrits en table 4 ; nous incluons aussi les scores BLEU obtenus sur l'ensemble de validation par les modèles de génération correspondants à fins

9. Rappelons cependant que l'algorithme FastText encode linéairement les régularités orthographiques, ce qui améliore ses résultats sur la tâche d'analogie formelle : le jeu d'analogies de Grave *et al.* (2018) inclut notamment des alternances flexionnelles, qui utilisent souvent des affixes orthographiquement réguliers.

Plongements	Validation	Meilleur ex. d'usage	Meilleur *definiendum*
aléatoires	19,80	17,10	16,20
word2vec	31,60	17,50	17,80
GloVe	28,40	17,60	18,00
FastText MC	30,30	17,80	18,70
FastText FB	32,30	17,30	18,80

TABLE 4: Génération de définitions : lignes de référence (scores BLEU).

de comparaisons. Premièrement, nous voyons que la marge de différence entre les représentations aléatoires et les jeux de plongements lexicaux est bien moindre que ce qu'on observe pour les modèles entraînés. Ceci s'explique en partie du fait qu'aucune des définitions pour ces lignes de référence n'est apprise : toute représentation même aléatoire est associée à une production tirée de l'ensemble de validation, et par conséquent toutes sont stylistiquement parfaites en ce qu'elles réemploient les tournures exactes du jeu d'entraînement. Par conséquent on peut supposer que tout écart à la référence s'explique par une différence d'ordre sémantique, plutôt que stylistique. Admettons donc que l'écart des scores BLEU issus de ces lignes de référence et de nos modèles reflète l'importance de la similarité sémantique : alors, bien qu'on puisse supposer que cette similarité sémantique est encodée dans nos plongements (ce que l'on peut voir en comparant les gains importants des modèles appris sur des plongements aux gains pauvres du modèle appris sur des représentations aléatoires), elle ne l'est pas aussi directement que la dépendance linéaire mesurée en analogie formelle (ce qui s'observe par l'écart limité entre les vecteurs aléatoires et des plongements dans ces lignes de référence).

De plus, si l'on observe les scores BLEU obtenus en récupérant le *definiendum* le plus similaire dans l'ensemble d'entraînement, on voit que les jeux de plongements distributionnels où la similarité entre *definienda* induit une plus grande similarité entre définitions sont ceux qui produisent les meilleurs résultats sur la tâche de génération de définitions. Ceci tend à confirmer que la génération de définitions requiert que le vecteur passé en entrée encode la similarité sémantique ; par extension, ceci confirmerait de plus l'utilité de cette tâche pour l'évaluation des jeux de plongements lexicaux.

Notons cependant que d'autres explications peuvent répondre à ces faits. En particulier les plongements lexicaux non-contextualisés représentent généralement tous les sens associés à un mot-type par le même vecteur : [10] par conséquent les phénomènes de poysémie et d'homonymie perturbent les mesures de similarité sémantique, et à son tour ceci implique que le *definiendum* le plus 'similaire' peut correspondre à un sens autre que celui visé par la définition auquel on l'associe. Enfin, la métrique BLEU employée ici peut n'être pas adaptée à l'évaluation que nous conduisons.

6 Analyse d'erreurs

Jusqu'ici, nous avons analysé les résultats produits par nos modèles à l'aide des scores BLEU et de la perplexité. Ces métriques peinent cependant à capturer la composante sémantique des productions d'un modèle ; notre emploi ici se justifie essentiellement par l'utilisation de ces métriques dans la littérature existante. Nous remarquons que la difficulté inhérente à la génération de définitions ne réside non pas dans des questions de variation ou cohérence stylistique mais bien plutôt dans

10. En particulier, on s'attend à ce que le sens le plus fréquent domine dans la représentation vectorielle, cf. Arora *et al.* (2018); Bartunov *et al.* (2016).

l'ancrage sémantique et la véracité du contenu des définitions : une définition stylistiquement parfaite et textuellement très proche de la référence peut être entièrement erronnée et inutilisable.

À titre d'illustration, envisageons une définition du mot "*chimique*" qui serait "*Relatif à ou issu de la Martinique*" plutôt que "*Relatif à ou issu de la chimie*", comme ce mot est définit dans le wikitionnaire : les métriques textuelles automatiques telles que BLEU ne pénalisent pas particulièrement ce type d'exemples — du moins, cet exemple erronné sera préféré presque systématiquement à une définition utilisant une autre formulation, mais cependant valide, telle que par exemple celle du TLFi : [11] "*Qui par nature appartient à la chimie, qui relève de la chimie.*".

De fait, cette illustration est loin d'être invraisemblable. Ayant sélectionné aléatoirement un échantillon E_{GLAWI} de 10000 définitions de GLAWI, nous avons calculé pour chaque définition d_i la distance d'édition minimale au reste de l'échantillon, c'est-à-dire :

$$d_i = \min\{D_{edit}(d_i, d_j) \quad \textit{tel que} \quad d_j \in E_{\text{GLAWI}} \ \wedge \ d_j \neq d_i\}$$

où D_{edit} correspond à la distance d'édition (ou de Levenshtein), définie sur les mots pluôt que les caractères, et d_i et d_j sont des définitions de l'échantillon E_{GLAWI}. Nous avons pu observer que 77,19% des définitions de notre échantillon ne différaient que par un mot ajouté, supprimé ou remplacé d'une autre définition de l'échantillon. De manière générale, les dictionnaires utilisent avec une grande fréquence un inventaire limité de tournures spécifiques, rendant la question de l'évaluation par métriques automatiques d'autant plus prégnante.

En bref, si le problème de la paraphrase est commun à toute tâche de génération automatique, il se présente de manière encore plus épineuse dans le cadre de la génération de définitions de par l'effet conjoint de la relative pauvreté des tournures stylistiques employées et de la proéminence de l'ancrage sémantique dans ce qui constitue une bonne ou une mauvaise définition. C'est aussi cette importance primordiale de l'ancrage sémantique, paradoxalement, qui fait l'attrait de la tâche générative pour l'évaluation des plongements lexicaux.

D'autres métriques similaires à BLEU pourraient être envisagées, notamment METEOR (Banerjee & Lavie, 2005; Elloumi *et al.*, 2015, pour l'adaptation au français) ; [12] nous notons toutefois que ces métriques ne ciblent pas spécifiquement les éléments lexicaux clefs ("*Martinique*" ou "*chimie*" dans notre exemple), mais prennent également en compte les éléments purement stylistiques dans le calcul d'un score : aussi le problème que nous soulignons n'est pas résolu par de telles métriques.

Afin de pallier ce défaut et d'obtenir une compréhension plus détaillée des productions qu'ils génèrent, nous sélectionnons aléatoirement 100 exemples et comptabilisons (i) le nombre de productions valides pour une partie du discours autre que celle du *definiendum*, (ii) le nombre de productions où le *definiendum* est présent dans sa propre définition et (iii) le nombre de productions contenant un mot ou un syntagme répété. Ces informations supplémentaires nous permettent d'établir une vision plus fine du genre d'erreurs de génération commises par nos modèles ; comme souligné plus haut, une définition peut évidemment être entièrement erronnée tout en n'enfreignant aucun de ces critères.

Les résultats correspondants sont consignés dans la table 5. Si, à première vue, les représentations aléatoires semblent autant erronées que les plongements GloVe, nous notons que les erreurs des

11. Trésor de la Langue Française informatisé. Disponible à l'adresse suivante : http://atilf.atilf.fr/.

12. Dans le cas de Elloumi *et al.* (2015), la ressource lexicale exploitée afin d'établir des relations de synonymies recouvre une partie significative de notre jeu de données, car les deux sont dérivés du wiktionnaire. Si nous employions cette métrique, des exemples tirés du jeu de test pourraient par conséquent être pris en compte dans le calcul de scores sur les jeux d'entraînement et de validation.

Plongements	Mauvaise POS	Auto-référence	Répétitions
aléatoires	25	1	6
word2vec	19	7	4
GloVe	24	2	5
FastText MC	16	7	5
FastText FB	22	4	0

TABLE 5: Erreurs typiques de génération.

vecteurs aléatoires sont souvent de plus grande ampleur : par exemple le nombre de répétitions dans une seule définition y est plus important. Un autre défi majeur rencontré par tous nos modèles est de distinguer les différentes parties du discours, ce que l'on peut supposer être dû à l'usage d'un unique encodeur pour le *definiendum* et l'exemple d'usage. De manière intéressante, les modèles basés sur des représentations non-aléatoires produisent davantage d'auto-références, et ce en l'absence de tout mécanisme de copie. Ceci peut être directement imputé à ce que l'espace sémantique des vecteurs distributionnels est effectivement structuré d'une manière exploitable par le réseau de neurones artificiel : bien qu'il s'agisse à proprement parler d'une erreur, l'auto-référence nous indique que la représentation contextualisée du *definiendum* influence effectivement le processus de génération, en ce que le décodeur choisit d'émettre le symbole au sens le plus similaire à cette entrée.

Penchons-nous à présent davantage sur les facteurs sémantiques qui peuvent conduire à une définition erronnée. Ceux-ci sont difficiles à étudier de manière systématique, en partie du fait que les erreurs de génération peuvent impacter ces facteurs sémantiques (notamment dans le cas fréquent où la partie du discours de la définition générée ne correspond pas à celle du *definiendum*), en partie aussi du fait du faible ancrage sémantique de nos modèles, qui conduit souvent à des productions difficiles à juger sur certains critères. Nous étudions par conséquent trois critères pour lesquels on peut espérer une certaine rigueur : (i) si le champ sémantique des éléments lexicaux présents dans la production est en lien avec le sens ciblé par l'exemple d'usage ; (ii) si le champ sémantique des éléments lexicaux présents dans la production est en lien avec un quelconque sens du *definiendum* et (iii) la proportion de productions suivant une forme *genus–differentia* [13] où le *genus* est un hyperonyme du *definiendum*.

Plongements	Champ sémantique inapproprié		*Genus–differentia*		
	à la définition visée	à tout sens	# défs. concernées	# avec genus incorrect	*(Pourcentage)*
aléatoires	93	91	58	50	86,2 %
word2vec	63	56	62	35	56,5 %
GloVe	67	57	65	40	61,5 %
FastText MC	61	45	69	38	55,1 %
FastText FB	70	57	65	41	63,1 %

TABLE 6: Erreurs à caractère sémantique.

Les résultats correspondants sont consignés dans la table 6. Nous donnons les résultats pour chacun des trois aspects sémantiques discutés précédemment, ainsi que le nombre de définitions produites

13. Les définitions de forme *genus–differentium* sont composées d'un mot ou d'une expression donnant la classe générale sémantique, le *genus*, et d'une expression restreignant cette classe au sens visé par le *definiendum*. Noraset *et al.* (2017) remarquent que cette forme de définition est très fréquente : 85 % des définitions de WordNet (Fellbaum, 1998) et 50 % des définitions du GCIDE y correspondraient.

ayant une forme *genus–differentia*. Nous soulignons que les représentations aléatoires produisent fréquemment des définitions métalinguistiques ("*variante orthographique de ...*", "*synonyme de...*", etc.), ce que nous suggérons être dû à l'incapacité de ces modèles à lier cohéremment un *definiendum* à un possible hyperonyme. De manière plus générale, la vision d'ensemble qui se dégage de cette évaluation manuelle indique clairement que l'adéquation sémantique des définitions demeure un défi majeur. Un point intéressant est que les scores attribués aux représentations aléatoires sont encore une fois nettement supérieurs à ceux attribués aux autres jeux de plongements lexicaux. En cela les résultats suggèrent encore une fois que la génération de définitions peut dans une certaine mesure servir de crible pour les représentations distributionnelles, puisque les définitions issues d'entrées sémantiquement incohérentes paraissent clairement moins acceptables. Il est intéressant de remarquer qu'ils indiquent aussi que les plongements FastText MC seraient les plus performants.

1.	Entrée	la **structuration** de la pensée.
	Réf.	*Action de structurer, de donner une structure.*
	aléatoire	`qualité de ce qui est net.`
	word2vec	`action de composer.`
	GloVe	`action de grouper.`
	FastText MC	`action de grouper.`
	FastText FB	`action de structurer.`
2.	Entrée	**chercheur** d'or, **chercheur** d'aventures, **chercheur** de querelles.
	Réf.	*Personne qui cherche.*
	aléatoire	`étoffe de soie, de coton, etc.`
	word2vec	`spécialiste de l'étude des connaissances.`
	GloVe	`celui qui s'occupe de statistique.`
	FastText MC	`celui, celle qui connaît.`
	FastText FB	`celui, celle qui connaît.`
3.	Entrée	le grand **mélinet** est une petite plante de la famille des bourraches qui se rencontre dans les terrains sablonneux relativement humides .
	Réf.	*Nom vulgaire de borraginées (cérinthe, tournefort), dont quelques espèces sont indigènes dans le midi.*
	aléatoire	`espèce d'insecte lépidoptère (papillon) de la famille des géométridés (noctuidae), dont les ailes sont caduques alternes.`
	word2vec	`plante monocotylédone de la famille des saxifragacées.`
	GloVe	`famille de plantes dicotylédones monopétales.`
	FastText MC	`qui croît sur les écorces.`
	FastText FB	`synonyme de préfloraison.`

TABLE 7: Productions choisies du jeu de validation (*definiendum* en gras dans l'entrée ; échantillon non-représentatif).

Parmi les erreurs à caractère sémantiques, d'autres que celles mentionnées dans la table 6 sont plus difficiles à annoter de manière systématique. La table 7 contient quelques exemples choisis associant une entrée à la définition attendue et aux productions de chacun des modèles. Plusieurs modèles produisent des définitions identiques pour des entrées identiques, ce qui suggère que les résultats en génération de définitions dépendent en grande partie des données lexicographiques utilisées. Certains des faits que nous mentionnons plus haut sont aussi visibles ici : par exemple, les productions sémantiquement inadéquates des représentations aléatoires ou bien la tendance générale des différents modèles à ignorer l'exemple d'usage pour se focaliser sur un sens particulier du *definiendum* (exemple 2). Enfin, un problème épineux concerne le cas de définitions en domaine de spécialité : sans une bonne connaissance de la botanique, il est difficile de juger de l'adéquation de la production du modèle word2vec pour l'exemple 3.

7 Conclusions

Cet article vise à adapter la tâche de génération de définitions à la langue française. À ce titre, nous proposons un jeu de données, des jeux de plongements normalisés et plusieurs éléments de comparaison pour les recherches ultérieures sur ce sujet. [14] Nos expériences suggèrent que la tâche de génération de définitions permet d'évaluer des représentations vectorielles car elle distingue proprement des représentations aléatoires de plongements lexicaux préentraînés. Les aspects qu'elle mettrait en exergue diffèrent de ceux capturés par l'analogie formelle : elle permet d'évaluer si des composantes sémantiques sont encodées de manière non linéaire dans les vecteurs, et favorise les jeux de plongements lexicaux où ces composantes structurent le plus clairement l'espace vectoriel.

Nous notons toutefois que la capacité à distinguer des représentations aléatoires de représentations sémantiquement cohérentes n'est qu'un des éléments nécessaires pour établir que cette tâche puisse effectivement jouer le rôle d'outil d'évaluation que la littérature suggère. D'autres expériences sont évidemment requises. En particulier, il serait utile de mettre en correspondance les productions générées par des modèles avec les jugements de locuteurs natifs quant à leur adéquation sémantique. Un autre point qu'il sera crucial d'étudier dans des travaux futurs consiste à établir des métriques automatiques et critères d'évaluations fiables adaptés à la tâche de génération de définitions.

L'évaluation manuelle que nous avons conduite laisse penser que ces modèles sont encore loin d'être effectivement utilisables. En vue d'améliorer ces résultats, plusieurs points méthodologiques demanderaient un examen plus minutieux : notamment les effets des choix d'hyperparamètres, tant pour les jeux de plongements que pour les modèles de génération. Le point central de la présente étude concernant l'adéquation de la génération de définitions à l'évaluation des plongements lexicaux, nous avons étudié en priorité comment différentes architectures de plongements interagissaient avec la tâche en question. Il est par conséquent plus que vraisemblable que les hyperparamètres optimaux pour la génération de définitions varient pour chaque jeu de plongements ; aussi l'établissement d'un protocole de séléction des hyperparamètres est un point méthodologique important que nous comptons aborder dans nos travaux futurs.

Ces travaux suggèrent aussi plusieurs pistes de recherches futures, en particulier sur le traitement des définitions en domaine de spécialité ou bien sur l'adaptation de la tâche à d'autres langues. Étendre la tâche de la génération de définitions dans un contexte multilingue nous paraît particulièrement crucial : non seulement parce qu'enrichir les jeux de données relatives à cette tâche permettra sans doute d'avancer nos connaissances en sémantique distributionnelles et représentations neuronales, mais aussi, et surtout, parce qu'une des applications pratiques majeures de la génération de définitions consiste en la documentation de langues peu dotées.

Remerciements

Nous remercions trois relecteurs anonymes dont les commentaires ont permis une amélioration significative du présent manuscrit. Ce travail a bénéficié d'une aide de l'État, gérée par l'Agence Nationale de la Recherche, au titre du projet Investissements d'Avenir Lorraine Université d'Excellence, portant la référence ANR-15-IDEX-04-LUE.

14. Ces éléments seront disponibles à l'adresse suivante : https://github.com/TimotheeMickus/dm-french

Références

ARORA S., LI Y., LIANG Y., MA T. & RISTESKI A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, **6**, 483–495. DOI : 10.1162/tacl_a_00034.

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.

BARTUNOV S., KONDRASHKIN D., OSOKIN A. & VETROV D. P. (2016). Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, p. 130–138.

BATCHKAROV M., KOBER T., REFFIN J., WEEDS J. & WEIR D. (2016). A critique of word similarity as a method for evaluating distributional semantic models. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, p. 7–12, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/W16-2502.

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

BOSC T. & VINCENT P. (2018). Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 1522–1532 : Association for Computational Linguistics.

CHANG T., CHI T., TSAI S. & CHEN Y. (2018). xSense : Learning Sense-Separated Sparse Representations and Textual Definitions for Explainable Word Sense Networks. arXiv : 1809.03348.

CHODOROW M. S., BYRD R. J. & HEIDORN G. E. (1985). Extracting semantic hierarchies from a large on-line dictionary. In *23rd Annual Meeting of the Association for Computational Linguistics*, p. 299–304, Chicago, Illinois, USA : Association for Computational Linguistics. DOI : 10.3115/981210.981247.

COAVOUX M. (2017). *Discontinuous Constituency Parsing of Morphologically Rich Languages*. Thèse de doctorat, Univ Paris Diderot, Sorbonne Paris Cité.

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.

ELLOUMI Z., BLANCHON H., SERASSET G. & BESACIER L. (2015). METEOR For Multiple Target Languages Using DBnary. In *MT Summit 2015*, Miami, United States. HAL : hal-01350109.

FELLBAUM C. (1998). *WordNet : An Electronic Lexical Database*. Bradford Books.

GADETSKY A., YAKUBOVSKIY I. & VETROV D. (2018). Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 266–271 : Association for Computational Linguistics.

GAUME B., HATHOUT N. & MULLER P. (2004). Word sense disambiguation using a dictionary for sense similarity measure. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA : Association for Computational Linguistics. DOI : 10.3115/1220355.1220528.

GAUME B., NAVARRO E., DESALLE Y. & GAILLARD B. (2014). Mesurer la similarité structurelle entre réseaux lexicaux. In *TALN-20 2014*, Proceedings of TALN-20 2014 : Atelier RLTLN, Réseaux Lexicaux et Traitement des Langues Naturelles, Marseille, France. HAL : hal-01321990.

GLADKOVA A., DROZD A. & MATSUOKA S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings : what works and what doesn't. In *SRWHLT-NAACL*.

GRAVE E., BOJANOWSKI P., GUPTA P., JOULIN A. & MIKOLOV T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

HATHOUT N. & SAJOUS F. (2016). Wiktionnaire's Wikicode GLAWIfied : a Workable French Machine-Readable Dictionary. In N. C. C. CHAIR), K. CHOUKRI, T. DECLERCK, M. GROBELNIK, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia : European Language Resources Association (ELRA).

HILL F., CHO K. & KORHONEN A. (2016a). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1367–1377, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1162.

HILL F., CHO K., KORHONEN A. & BENGIO Y. (2016b). Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, **4**, 17–30.

KUDO T. & RICHARDSON J. (2018). SentencePiece : A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 66–71, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-2012.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2019). FlauBERT : Unsupervised Language Model Pre-training for French. arXiv preprint : 1912.05372.

LENCI A. (2018). Distributional models of word meaning. *Annual review of Linguistics*, **4**, 151–171.

LEVY O. & GOLDBERG Y. (2014a). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, p. 171–180 : Association for Computational Linguistics. DOI : 10.3115/v1/W14-1618.

LEVY O. & GOLDBERG Y. (2014b). Neural word embedding as implicit matrix factorization. In Z. GHAHRAMANI, M. WELLING, C. CORTES, N. D. LAWRENCE & K. Q. WEINBERGER, Éds., *Advances in Neural Information Processing Systems 27*, p. 2177–2185. Curran Associates, Inc.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., VILLEMONTE DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2019). CamemBERT : a Tasty French Language Model. arXiv preprint : 1911.03894.

MICKUS T., PAPERNO D. & CONSTANT M. (2019). Mark my Word : A Sequence-to-Sequence Approach to Definition Modeling. *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*. HAL : hal-02362397.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space. arXiv : 1301.3781.

MIKOLOV T., YIH W.-T. & ZWEIG G. (2013b). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, p. 746–751.

NORASET T., LIANG C., BIRNBAUM L. & DOWNEY D. (2017). Definition modeling : Learning to define word embeddings in natural language. In *AAAI*.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, p. 311–318, USA : Association for Computational Linguistics. DOI : 10.3115/1073083.1073135.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

SCHÄFER R. (2015). Processing and querying large web corpora with the COW14 architecture. In P. BAŃSKI, H. BIBER, E. BREITENEDER, M. KUPIETZ, H. LÜNGEN & A. WITT, Éds., *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, Lancaster : UCREL IDS.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1715–1725, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/P16-1162.

SIERRA G., TORRES-MORENO J.-M. & MOLINA A. R. (2015). Regroupement sémantique de définitions en espagnol. arXiv : 1501.04920.

SWINGER N., DE-ARTEAGA M., IV N. T. H., LEISERSON M. D. M. & KALAI A. T. (2018). What are the biases in my word embedding ? arXiv preprint : 1812.08769.

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 30*, p. 5998–6008. Curran Associates, Inc.

YANG L., KONG C., CHEN Y., LIU Y., FAN Q. & YANG E. (2019). Incorporating sememes into chinese definition modeling. arXiv preprint : 1905.06512.

ZHANG H., DU Y., SUN J. & LI Q. (2019). Improving Interpretability of Word Embeddings by Generating Definition and Usage. arXiv preprint : 1912.05898.

ZHU R., NORASET T., LIU A., JIANG W. & DOWNEY D. (2019). Multi-sense Definition Modeling using Word Sense Decompositions. arXiv preprint : 1909.09483.