Étiquetage en parties du discours de langues peu dotées par spécialisation des plongements lexicaux

Pierre Magistry, Anne-Laure Ligozat, Sophie Rosset

▶ **To cite this version:**

Pierre Magistry, Anne-Laure Ligozat, Sophie Rosset. Étiquetage en parties du discours de langues peu dotées par spécialisation des plongements lexicaux. Conférence sur le Traitement Automatique des Langues Naturelles, May 2018, Rennes, France. hal-01793092

HAL Id: hal-01793092
https://hal.science/hal-01793092

Submitted on 23 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Étiquetage en parties du discours de langues peu dotées par spécialisation des plongements lexicaux

Pierre Magistry[1] Anne-Laure Ligozat[2] Sophie Rosset[1]
(1) LIMSI, CNRS, Université Paris-Saclay, Bât 508, rue John von Neumann, Campus Universitaire, F-91405 Orsay
(2) LIMSI, CNRS, ENSIIE, Université Paris-Saclay, Bât 508, rue John von Neumann, Campus Universitaire, F-91405 Orsay
{magistry,annlor,rosset}@limsi.fr

RÉSUMÉ
Cet article présente une nouvelle méthode d'étiquetage en parties du discours adaptée aux langues peu dotées : la définition du contexte utilisé pour construire les plongements lexicaux est adaptée à la tâche, et de nouveaux vecteurs sont créés pour les mots inconnus. Les expériences menées sur le picard, le malgache et l'alsacien montrent que cette méthode améliore l'état de l'art pour ces trois langues peu dotées.

ABSTRACT
POS tagging for low-resource languages by adapting word embeddings

This paper presents a new method for Part-of-speech tagging, adapted to low-resource languages : the context definition is adapted to the POS tagging task, and new vectors are created for unknown words. Experiments on Picard, Malagasy and Alsatian show that it improves the state of the art for all three languages.

MOTS-CLÉS : étiquetage en parties du discours, langues peu dotées.

KEYWORDS: POS tagging, low resource languages.

1 Introduction

De très grands corpus bruts et annotés sont désormais disponibles pour certaines langues, et les méthodes par apprentissage en Traitement automatique des langues s'appuient maintenant souvent sur leur existence : les corpus bruts permettent d'apprendre des plongements lexicaux, et les corpus annotés servent à l'apprentissage des modèles. Cependant, l'écrasante majorité des langues du monde ne dispose pas de telles ressources, et les méthodes existantes ne peuvent donc pas être utilisées sans adaptation.

En ce qui concerne l'étiquetage en parties du discours, le manque de ressources pour les langues peu dotées pose deux problèmes : la faible quantité de corpus bruts ne permet pas d'apprendre des plongements lexicaux de même qualité que pour les langues généralement étudiées ; par ailleurs, les données annotées sont également rares, et l'annotation de nouveaux corpus peut être difficile, notamment en raison de la difficulté à trouver des annotateurs. Plusieurs stratégies sont possibles pour pallier ces problèmes, par exemple s'appuyer sur des corpus parallèles ou sur des ressources de

langues proches. Cependant, nous proposons une autre stratégie, qui nous permet de nous situer dans le cadre le plus général possible : pas de corpus parallèle, pas de langue proche bien dotée.

Dans cet article, nous proposons donc une nouvelle méthode pour l'étiquetage en parties du discours de langues peu dotées, qui s'appuie sur une méthode de construction adaptée des plongements lexicaux. Cette méthode est indépendante de la langue, et a été testée sur des langues typologiquement distantes : deux langues régionales de France, l'alsacien et le picard, et une autre langue peu dotée, le malgache, une langue austronésienne.

2 État de l'art

Depuis quelques années, les architectures Bi-LSTM ont montré leur potentiel sur des tâches d'étiquetage de séquences, et en particulier d'étiquetage en parties du discours (Horsmann & Zesch, 2017). Cependant, ce type d'approche nécessite une très grande quantité de données, qui est loin des quantités disponibles dans le cas de langues peu dotées (15M de tokens pour entraîner les plongements lexicaux avec fastText et entre 50 000 et 2M de tokens annotés pour entraîner l'étiqueteur bi-LSTM). Peu d'évaluations ont été faites sur des langues peu dotées.

(Fang & Cohn, 2016) rapportent des résultats mitigés sur le malgache : ils utilisent un corpus aligné pour projeter les annotations depuis l'anglais et pour modifier le réseau de neurones afin de dépasser l'état de l'art, sans quoi les performances sont mauvaises.

Les performances de ces réseaux de neurones sont fortement dépendantes de la qualité des plongements lexicaux. L'un des modèles très largement utilisé est le modèle SkipGram, et en particulier l'outil fastText (Bojanowski *et al.*, 2016). Deux caractéristiques de fastText sont intéressantes dans le cadre de l'étiquetage en parties du discours de langues peu dotées : il prend en compte des informations internes aux mots, ce qui permet de capturer des informations morphologiques, essentielles pour l'étiquetage en parties du discours ; de plus il est capable de générer des vecteurs pour les mots hors vocabulaire, ce qui est également indispensable dans le cadre de langues peu dotées.

Concernant les langues étudiées, le cas du malgache a déjà été évoqué avec les travaux de (Fang & Cohn, 2016). Le picard et l'alsacien ont fait l'objet de travaux fondés sur une transposition de mots outils (Bernhard & Ligozat, 2013; Magistry *et al.*, 2017), et, pour l'alsacien, sur l'annotation collaborative de corpus (Millour *et al.*, 2017). Nous comparerons nos résultats aux méthodes par transposition et à notre implémentation d'un étiqueteur basé sur un modèle MaxEnt semblable à celui utilisé dans (Millour *et al.*, 2017).

3 Adaptation des plongements lexicaux au cas des langues peu dotées

Les plongements lexicaux sont souvent considérés comme un outil de représentation de la similarité syntactico-sémantique, mais en réalité, étant censés capturer des similarités distributionnelles, ils peuvent être adaptés à différents niveaux d'analyse linguistique. Une telle spécialisation apparaît même nécessaire lorsque ceux-ci sont entraînés sur de petites quantités de données afin que les similarités puissent être plus facilement capturées.

3.1 Description du système

Dans cet article, nous nous appuyons sur une architecture classique de Bi-LSTM, pour laquelle nous utilisons l'implémentation YASET (Tourille *et al.*, 2017), reprenant l'architecture de (Lample *et al.*, 2016).

Nous avons fixé les paramètres de YASET en tenant compte du fait que nos corpus sont petits et donc du risque de sur-apprentissage. Nous avons également essayé d'utiliser hyperopt mais étant donné la variété des situations étudiées, il est peu probable de trouver un jeu de paramètres qui conviennent à toutes. Nous utilisons une couche cachée de taille 30, et optimisons par *adam* avec un loss ratio de $0,001$ et un *dropout* à $0,5$.

3.2 Plongements lexicaux pour l'étiquetage en parties du discours

Dans ce travail, nous proposons une définition du contexte pour les plongements lexicaux qui est fondée sur la tâche d'étiquetage visée. Nous utilisons le modèle SkipGram avec un échantillonnage de contre-exemples (*negative sampling*) popularisé par *word2vec* (Mikolov *et al.*, 2013), et étendu ensuite dans fastText (Bojanowski *et al.*, 2016) pour prendre en compte des informations internes aux mots, en représentant les mots par leurs n-grammes de caractères.

Le système fastText cherche à prédire les mots du contexte étant donné un mot cible ou un n-gramme de caractères, pour lequel on cherche à construire un vecteur. fastText est ici utilisé comme système baseline pour construire les plongements lexicaux (voir figure 1b).

Nous proposons d'étoffer ce modèle en ne nous limitant pas aux formes avoisinantes pour définir les éléments du contexte afin de construire des plongements lexicaux spécialisés pour la tâche d'analyse en parties du discours ; ces plongements seront appelés MorphoSyntactic Embeddings (MSE).

Afin de spécialiser les plongements lexicaux, nous forçons le modèle SkipGram à s'intéresser aux indices généralement suivis pour une telle analyse (y compris lorsque celle-ci est manuelle). Nous construisons ainsi un modèle qui vise à prédire ces indices à partir d'un mot cible donné.

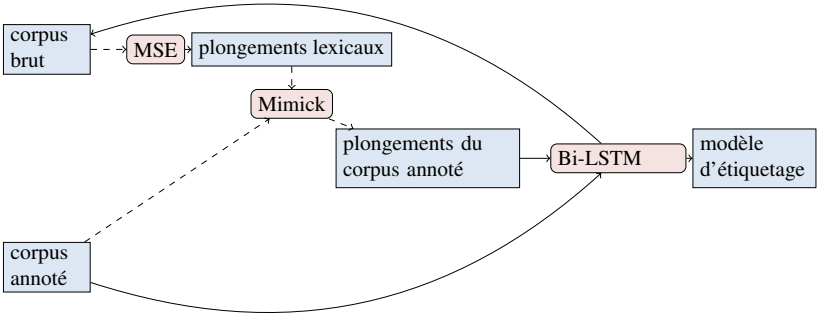
Les indices retenus sont les suivants :

— formes précédente et suivante ;
— morphèmes du mot cible ;
— morphèmes des mots précédent et suivant ;
— parties du discours des mots précédent et suivant ;
— mots grammaticaux les plus proches à droite et à gauche du mot cible.

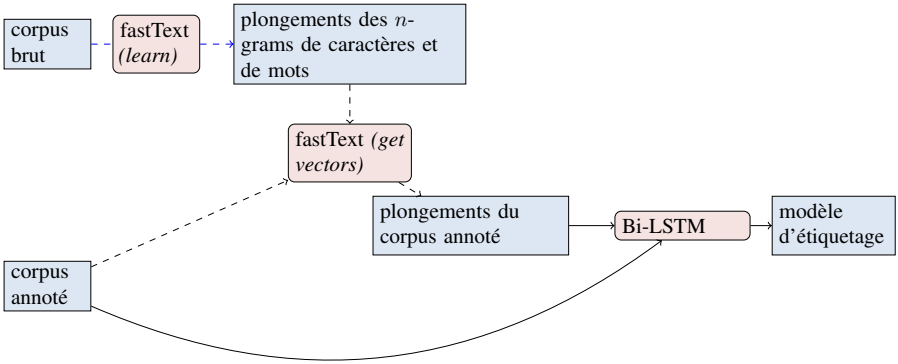
Pour obtenir les morphèmes, nous utilisons l'outil Morfessor (Virpioja *et al.*, 2013) [1]. Afin d'obtenir les informations sur les étiquettes des mots voisins et les mots grammaticaux, nous procédons de manière itérative (voir figure 1a). Une première version de notre système ignorant ces informations est utilisée pour annoter le corpus brut, et cette information est prise en compte lors d'une seconde exécution.

Un exemple d'indices considérés pour une occurrence de *Làcha* dans notre corpus alsacien est donné à la figure 2 : la première ligne correspond au mot courant avec son contexte, la seconde ligne est

1. Cet outil segmente la forme et produit donc un sous-ensemble des n-grammes qu'elle contient. En nous limitant à ce sous-ensemble, nous pensons éviter une partie du bruit qui nuit au modèle de fastText, particulièrement dans le cas de nos petits volumes de données. De plus, à la différence de fastText nous prenons en compte la morphologie des mots voisins.



(a) MSE



(b) fastText

FIGURE 1: Architectures des systèmes : avec plongements MSE et avec fastTest (les pointillés représentent des apprentissages non supervisés)

Èbbis	genschtigeres	às	Làcha	gìtt	’s	uf	dr	gànza	Walt	nìt
PRON	ADJ	SCONJ	?	VERB	PRON	ADP	DET	ADJ	NOUN	PART
ebb-is	gen-scht-iger-es	às	Làch-a	gìt-t	’s	uf	fr	gànz-a	Walt	nìt
quelque chose	plus abordable	que	rire	exister	ça	sur	le	entier	monde	pas

indices pour ‘Làcha’			
Context type	value	Context type	value
morpheme	Làch-	next-morph	gìt-
morpheme	-a	next-morph	-t
prev-form	às	next-form	gìtt
prev-tag	SCONJ	next-tag	VERB
prev-funct-word	às	next-funct-word	’s

FIGURE 2: Extraction des indices morphosyntaxiques

la séquence de parties du discours qui ont été attribuées lors de la première annotation, la troisième ligne correspond aux morphèmes de chaque mot, déterminés par *Morfessor*, et la quatrième ligne est la traduction française. À partir de ces informations, les indices extraits pour cette occurrence du mot *Làcha* sont indiqués dans le tableau : par exemple, les morphemes *Làch-* et *-a* du mot cible, la forme précédente *às* etc.

3.3 Gestion des mots inconnus

L’une des difficultés dans l’utilisation de ces modèles est la prise en compte des mots peu fréquents, voire absents du corpus d’apprentissage. Cette difficulté est d’autant plus importante lorsque les corpus disponibles sont de faible taille : lorsque les mots hors vocabulaire représentent environ la moitié des mots (ce qui est le cas dans les corpus étudiés), il n’est pas possible de leur attribuer la même représentation à tous.

Dans cet article, nous nous comparons à fastText comme méthode de base. Celui-ci crée des vecteurs pour les mots inconnus en additionnant les vecteurs qu’il a construit pour des sous-mots (n-grammes de caractères). Notre système ne construit pas de représentation pour les n-grammes, mais uniquement pour les mots cibles observés dans le corpus de données brutes. Pour générer des vecteurs pour les mots inconnus, nous utilisons le système proposé par (Pinter *et al.*, 2017), Mimick, qui entraîne un Bi-LSTM sur les caractères pour prédire des vecteurs en fonction de la graphie des mots. Mimick est entraîné sur les vecteurs déjà existants. L’un des paramètres essentiels dans cette configuration est le nombre minimal d’occurrences à partir duquel un token sera pris en compte pour le calcul des plongements lexicaux : en effet, les fréquences étant globalement assez faibles dans nos corpus, fixer un seuil trop haut éliminerait trop de mots (qui seraient cependant gérés par Mimick, mais Mimick a besoin de suffisamment de vecteurs également) ; en revanche, fixer un seuil trop bas conduit à des plongements de mauvaise qualité.

4 Validation sur le français

Nous avons tout d’abord mené des tests sur le français car il existe de nombreuses ressources pour le français, ce qui permet de contrôler la quantité de données utilisée à la fois pour créer les plongements lexicaux et pour l’entraînement de l’étiqueteur.

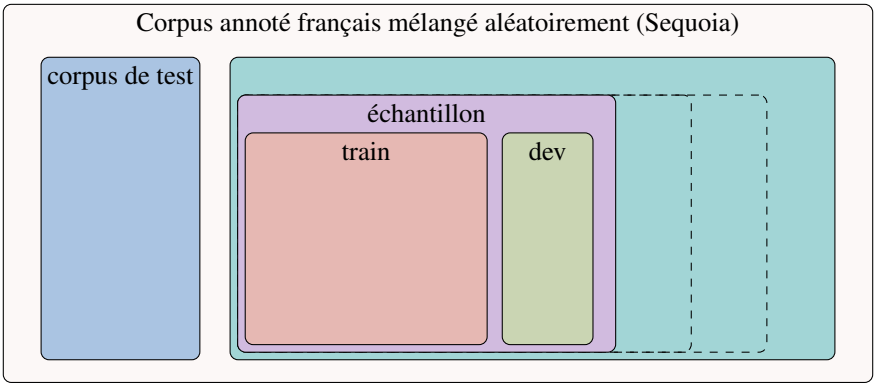


FIGURE 3: Échantillonnage des données annotées. Les tailles des corpus annotés varient entre 500 et $50,000$ tokens. Le corpus de test a été mis de côté pour servir de jeu d'évaluation pour tous les tests.

Pour ces expériences, nous avons utilisé comme données brutes un corpus de la Wikipédia française au format texte [2] et comme données annotées le corpus Sequoia (Candito & Seddah, 2012), qui a été converti vers le jeu d'étiquettes Universal POS tags.

Dans un premier temps, nous avons étudié l'influence de la quantité de données utilisée sur la qualité de l'étiquetage. Pour cela, nous avons tout d'abord mis de côté 20% du corpus Sequoia, que nous utilisons comme test pour toutes les expériences. Pour les corpus d'entraînement, nous avons mélangé les phrases et nous vérifions que les plus gros corpus incluent toujours les plus petits afin de limiter les effets de genre ou de thème (voir figure 3).

Nous avons utilisé des tailles de corpus brut allant de 200 000 tokens à 20M pour les plongements lexicaux, et de 500 à 50 000 tokens pour les données annotées. Nous évaluons notre système sur ces différentes tailles de corpus, et comparons notre méthode de construction de plongements à fastText. Les résultats sont donnés en figure 4.

Ces résultats montrent que les performances sont très bonnes lorsque les quantités de données sont importantes, à la fois en terme de corpus brut et de corpus annoté. Néanmoins, dans le cas, artificiel pour le français, de faibles volumes de données (en bas à gauche des matrices), notre système obtient de meilleurs résultats que fastText.

4.1 Plongements lexicaux

L'analyse des plongements lexicaux pour 500 000 tokens permet d'explorer le comportement des plongements crées par fastText et MSE. La figure 5 montre ces plongements après réduction de dimensionnalité à 2 avec l'outil T-SNE. Chaque point représente une forme du corpus et les couleurs représentent la partie du discours la plus souvent attribuée à cette forme. Dans la partie droite de la figure, qui correspond à nos plongements lexicaux, les parties du discours sont relativement séparées les unes des autres, contrairement à ce qui est observé avec les plongements lexicaux de fastText. C'est une explication possible de la différence de score de YASET lorsqu'il est utilisé avec l'une ou l'autre des représentations.

2. Mis à disposition par `https://ufal.mff.cuni.cz/w2c`

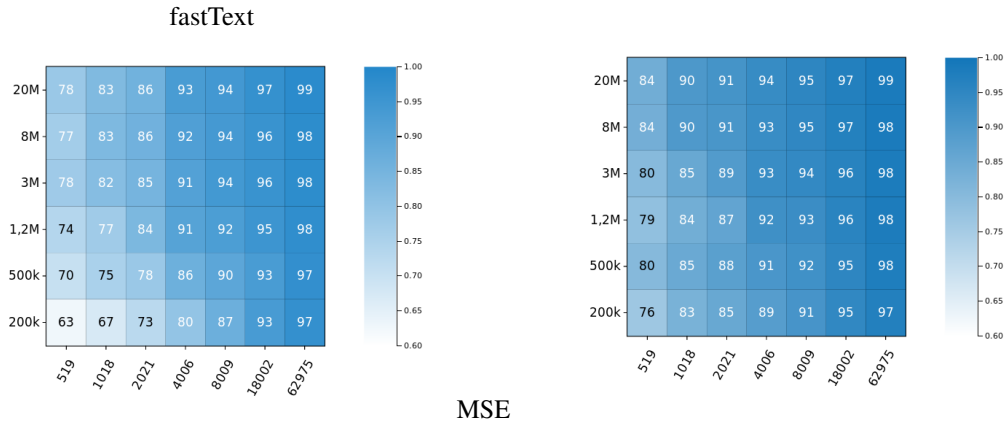


FIGURE 4: Scores d'étiquetage en parties du discours obtenus sur le corpus Sequoia en utilisant fastText ou nos embeddings (MSE), en faisant varier la taille du corpus annoté (abscisse) et du corpus brut (ordonnée)

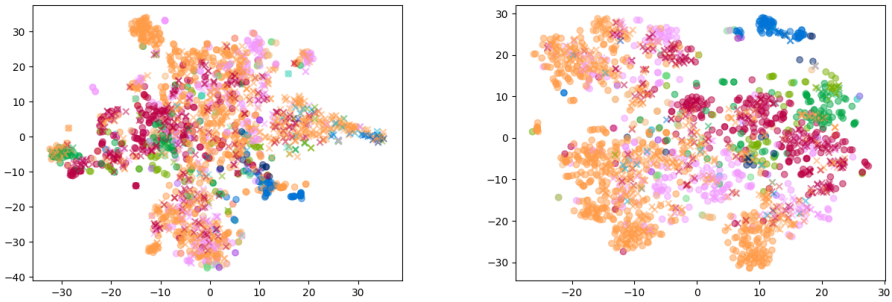


FIGURE 5: Plongements lexicaux obtenus avec fastText et avec MSE. Chaque point représente un mot, et chaque couleur une partie du discours

5 Expériences sur des langues peu dotées

Notre objectif étant de créer une méthode d'étiquetage en parties du discours pour des langues peu dotées, nous avons testé notre système sur trois langues peu dotées : deux langues régionales de France, le picard et l'alsacien, et le malgache, qui nous permet de comparer nos travaux à un état de l'art récent.

	données		exactitude			
langue	brutes	annotées	MSE	fastText	transposition	MaxEnt
alsacien	200 000	12 600	**0,91**	0,86	0,78	0,85
picard	1,9M	9 640	**0,89**	0,82	0,71	0,78
malgache	2M	4 230	**0,91**	0,84	n/a	0,86
malgache (Fang & Cohn 2016)			0,87			

TABLE 1: Scores d'étiquetage en parties du discours

5.1 Langues étudiées

Le malgache est une langue de la famille des langues austronésiennes. C'est une langue officielle à Madagascar où elle est parlée par plus de 20M de personnes. Il s'agit d'une langue de type VOS, morphologiquement riche, de type agglutinative.

L'alsacien est une langue parlée dans le nord est de la France, principalement en Alsace. Elle est composée de plusieurs dialectes, qui pour la plupart sont issus des langues alémaniques et franciques. Si l'alsacien présente de nombreuses variantes inter- et intra-dialectales, les principales caractéristiques morphologiques se retrouvent dans toutes ces variantes. Pour l'essentiel, on peut retenir que les verbes reçoivent des morphèmes de temps, mode et nombre et les substantifs des morphèmes de nombre, cas et genre.

Le picard est une langue d'oïl, qui appartient à la famille des langues romane. La zone géographique du picard couvre la région des Hauts-de-France et la province de Hainaut en Belgique. Bien que proche du français, le picard présente des particularités. Ainsi, l'ordre des mots peut être différent. Par exemple, *il o foait keud assé* se traduit en *il fait assez chaud* où l'on peut constater que une inversion de l'ordre entre l'adverbe (assé/assez) et l'adjectif (keud/chaud). De plus le picard présente de nombreuses contractions de prépositions et déterminants (par exemple *d'ches, su, al*) et des néologismes construits par composition.

5.2 Corpus

Les corpus bruts utilisés ont des tailles de l'ordre de 2 millions de tokens pour le malgache et le picard, et de 200 000 tokens pour l'alsacien (voir tableau 1). Le corpus malgache est composé d'articles du site *Global Voice* [3], également utilisés par (Fang & Cohn, 2016). Pour le picard, la base textuelle Picartext constitue la source des données brutes, et pour l'alsacien, ce sont les articles de la Wikipédia alémanique annotés comme étant en alsacien [4].

En ce qui concerne les données annotées les corpus du projet RESTAURE ont été utilisés [5]. Le corpus annoté pour l'alsacien contient environ 12 600 tokens, pour le picard environ 9 700 tokens, tandis que celui pour le malgache est à environ 4 200. Ces situations correspondent environ au milieu des matrices de la figure 4. Cependant, les deux langues régionales de France étudiées sont moins normalisées que le français, donc on peut s'attendre à de moins bons scores avec une méthode standard.

3. https://www.cs.cmu.edu/~ark/global-voices/
4. La Wikipédia alémanique contient des articles dans plusieurs dialectes correspondant à l'aire linguistique alémanique
5. https ://zenodo.org/communities/restaure

Tous les corpus annotés utilisent les Universal POS tags [6].

Pour le malgache, nous avons gardé la division train/test de (Fang & Cohn, 2016). Pour les deux langues régionales de France, les expériences ont été menées en validation croisée en 5 tirages après mélange aléatoire des phrases du corpus.

5.3 Résultats

Les résultats obtenus avec les différents systèmes sont présentés dans le tableau 1.

Les scores obtenus avec notre système avec les plongements MSE sont systématiquement supérieurs à ceux obtenus en utilisant fastText : 0,91 contre 0,86 en alsacien ; 0,89 contre 0,82 en picard ; et 0,91 contre 0,84 en malgache, ce qui nous situe bien au-dessus de l'état de l'art de (Fang & Cohn, 2016), sans utiliser de données bilingues.

Nous avons également indiqué les scores obtenus par une méthode de transposition proche de celle de (Bernhard & Ligozat, 2013), qui se fonde sur des étiqueteurs de langues proches : modèle allemand du Stanford POS Tagger (Toutanova *et al.*, 2003) pour l'alsacien, modèle MaxEnt français entraîné sur Sequoia pour le picard ; le malgache ne possède pas à notre connaissance de langue proche bien dotée. Cette méthode de transposition obtient des résultats bien inférieurs, bien qu'au-dessus de ceux de fastText en alsacien.

Enfin, la dernière colonne donne les scores d'un système très proche de MElt (Denis & Sagot, 2009), utilisé par (Millour *et al.*, 2017), et dont les performances sont également significativement en-dessous de notre système.

6 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode d'étiquetage en parties du discours pour les langues peu dotées, fondée sur une adaptation des plongements lexicaux. Les résultats obtenus dépassent l'état de l'art pour chaque des langues étudiées. La prise en compte des mots hors vocabulaire pourrait cependant être encore améliorée, notamment en prenant en compte leur contexte pour générer leur vecteur.

Remerciements

Ces travaux ont bénéficié du soutien de l'ANR (projet RESTAURE - référence ANR-14-CE24-0003).

6. http://universaldependencies.org

Références

BERNHARD D. & LIGOZAT A.-L. (2013). Hassle-free POS-Tagging for the Alsatian Dialects. In *Special volume on 'Non-Standard Data Sources in Corpus Based-Research'*.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2016). Enriching Word Vectors with Subword Information. *arXiv preprint ArXiv :1607.04606*.

CANDITO M. & SEDDAH D. (2012). Le corpus sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus : Syntactic annotation and use for a parser lexical domain adaptation method) [in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*, p. 321–334 : ATALA/AFCP.

DENIS P. & SAGOT B. (2009). Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, volume 1.

FANG M. & COHN T. (2016). Learning when to trust distant supervision : An application to low-resource POS tagging using cross-lingual projection. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 201 6*, p. 178–186.

HORSMANN T. & ZESCH T. (2017). Do LSTMs really work so well for PoS tagging ?–A replication study. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 727–736.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.

MAGISTRY P., LIGOZAT A.-L. & ROSSET S. (2017). Expériences d'étiquetage morphosyntaxique dans le cadre du projet RESTAURE. In *Atelier Diversité linguistique et TAL (DiLiTAL 2017) associé à la conférence TALN*.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, p. 3111–3119.

MILLOUR A., FORT K., BERNHARD D. & STEIBLE L. (2017). Toward a lightweight solution to the language resources bottleneck issue : creating a POS tagger for Alsatian using voluntary crowdsourcing . In *Traitement Automatique des Langues Naturelles (TALN)*, Orléans, France.

PINTER Y., GUTHRIE R. & EISENSTEIN J. (2017). Mimicking Word Embeddings using Subword RNNs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 102–112.

TOURILLE J., FERRET O., NÉVÉOL A. & TANNIER X. (2017). Neural architecture for temporal relation extraction : A bi-lstm approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 224–230, Vancouver, Canada : Association for Computational Linguistics.

TOUTANOVA K., KLEIN D., MANNING C. D. & SINGER Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, p. 173–180 : Association for Computational Linguistics.

VIRPIOJA S., SMIT P., GRÖNROOS S.-A. & KURIMO M. (2013). *Morfessor 2.0 : Python Implementation and Extensions for Morfessor Baseline.* Rapport interne, Helsinki.