La séparation des composantes lexicale et flexionnelle des vecteurs de mots

François Lareau Gabriel Bernier-Colborne Patrick Drouin
OLST, Université de Montréal, C.P. 6128, succ. Centre-Ville, Montréal QC H3C 3J7, Canada
{ francois.lareau | gabriel.bernier-colborne | patrick.drouin }@umontreal.ca

Résumé. En sémantique distributionnelle, le sens des mots est modélisé par des vecteurs qui représentent leur distribution en corpus. Les modèles étant souvent calculés sur des corpus sans pré-traitement linguistique poussé, ils ne permettent pas de rendre bien compte de la compositionnalité morphologique des mots-formes. Nous proposons une méthode pour décomposer les vecteurs de mots en vecteurs lexicaux et flexionnels.

Abstract.

Separating the lexical and grammatical components of semantic vectors
In distributional semantics, the meaning of words is modelled by vectors that represent their distribution in a corpus. Vectorial models being often built from corpora with little linguistic pre-treatment, they do not represent very well the morphological compositionality of words. We propose here a method to decompose semantic vectors into lexical and inflectional vectors.

Mots-clés : Sémantique distributionnelle ; compositionnalité ; flexion.

Keywords: Distributional semantics ; compositionality ; inflection.

1 Introduction

En sémantique distributionnelle, le sens des mots est représenté par des vecteurs qui représentent leur distribution en corpus (Turney & Pantel, 2010). Les modèles étant souvent calculés sur des corpus sans pré-traitement linguistique poussé, ils ne permettent pas de rendre bien compte de la compositionnalité morphologique des mots-formes. Cela mène à des aberrations. Par exemple, dans le modèle de Mikolov *et al.* (2013b), les vecteurs des quasi-synonymes *seems* et *appears* sont plus similaires que ceux de *seems* et *seemed*, qui appartiennent pourtant au même vocable (cf. table 1). Puisque ces vecteurs sont construits en discours, ils contiennent à la fois de l'information sémantique et de l'information d'autre nature tenant plus au fonctionnement des mots dans les phrases et aux propriétés des mots eux-mêmes qu'à leur sens comme tel. Ce bruit ne gêne généralement pas les recherches qui visent une approximation du sens des formes, mais peut gêner considérablement les travaux de nature purement linguistique. Nous cherchons, dans cet article, à éliminer le bruit morphosyntaxique contenu dans les vecteurs dans le but d'isoler le contenu sémantique.

seemed	*seems*	0,721
appeared	*appears*	0,657
seemed	*appeared*	0,723
seems	*appears*	0,814

TABLE 1 – Aberrations

Notre hypothèse est que, dans le vecteur d'une forme fléchie, on peut en isoler la partie lexicale de sa partie flexionnelle. Par exemple, comme l'illustre la figure 1, les vecteurs des verbes au passé *got*, *began* et *wrote* devraient pouvoir être décomposés en un vecteur lexical (GET, BEGIN ou WRITE) et un vecteur flexionnel représentant le passé (VBD [1]) qui est commun à ces formes. De la même façon, les formes *wrote, writes* et *write* [2] partagent un vecteur lexical commun qui représente le vocable WRITE.

Notre objectif est donc d'identifier automatiquement, dans des vecteurs construits *a priori*, des sous-vecteurs qui représentent le contenu flexionnel *vs* lexical. Dans l'exemple qui précède, nous cherchons donc à isoler le vecteur VBD à partir d'un ensemble d'observations effectuées sur des formes au passé (colonne de gauche dans la figure 1). Une fois ce vecteur

1. Nous utilisons les codes du *Penn Treebank*.
2. Même si *write* n'a pas de marqueur morphologique explicite, nous le considérons comme une forme fléchie puisqu'il porte de l'information grammaticale (infinitif, impératif ou présent de l'indicatif).

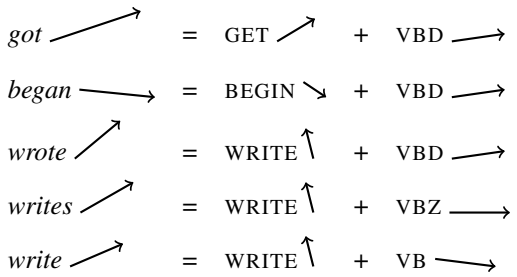

FIGURE 1 – Décomposition de vecteurs de mots en vecteurs lexicaux et flexionnels

isolé, nous pourrons le soustraire de la représentation vectorielle de *wrote* pour isoler le vecteur lexical épuré. En répétant le processus sur des formes à la troisième personne du singulier du présent de l'indicatif ou sur des formes nues, nous croyons qu'il sera possible d'isoler les vecteurs correspondants, qui devraient être relativement proches de ceux isolés à partir des formes au passé. L'intérêt d'une démarche qui rendrait possible l'identification et le retrait d'un vecteur flexionnel n'est pas négligeable puisqu'elle permettrait de maximiser la proximité sémantique entre les formes fléchies des mots et d'identifier les sens lexicaux « purs ».

2 Travaux antérieurs

La méthode que nous décrivons repose sur des représentations vectorielles de mots, qui peuvent être construites de différentes façons ; ici, nous utilisons des représentations apprises au moyen du modèle de langue neuronal *word2vec* (Mikolov *et al.*, 2013a,b). Des modèles de ce type ont été exploités dans de nombreuses applications du TAL, telles que l'étiquetage morphosyntaxique et sémantique, la segmentation, la reconnaissance automatique de la parole et la traduction automatique (Schwenk, 2007; Collobert *et al.*, 2011; Do *et al.*, 2014). Ces modèles apprennent des représentations distribuées de mots qui modélisent leurs propriétés sémantiques et morphosyntaxiques. Ces représentations peuvent notamment être exploitées afin d'estimer la similarité des mots, en utilisant une mesure de similarité de vecteurs telle que le cosinus.

La capacité des modèles de langue neuronaux à modéliser des régularités sémantiques et morphosyntaxiques a été démontrée au moyen de tâches de résolution d'analogies telles que « *homme* est à *roi* ce que *femme* est à __ ». Mikolov *et al.* (2013c) ont montré qu'il est possible de résoudre de telles analogies au moyen d'un modèle comme *word2vec* en appliquant des opérations simples aux représentations de mots, en l'occurrence en soustrayant le vecteur du mot *homme* de celui de *roi*, puis en additionant le vecteur de *femme*, le vecteur résultant ayant comme plus proche voisin *reine*. Les auteurs ont évalué leur modèle de langue sur des analogies sémantiques telles que *homme* : *roi* :: *femme* : *reine*, mais ont aussi créé à des fins d'évaluation un ensemble d'analogies morphosyntaxiques telles que *pomme* : *pommes* :: *voiture* : *voitures*. D'autres, comme Lazaridou *et al.* (2013) se sont intéressés à la compositionnalité des mots morphologiquement dérivés. Toutefois, à notre connaissance, personne n'a encore cherché à décomposer les vecteurs de mots pour en isoler les composantes lexicale et flexionnelle.

3 Données et méthodologie

Nous avons choisi de travailler sur les verbes en anglais, et ce, pour deux raisons. D'abord, à cause de la disponibilité d'un très gros modèle pré-entraîné, celui de Mikolov *et al.* (2013b). Ce modèle à 300 dimensions a été entraîné sur 100 milliards de mots du corpus *Google News* à l'aide du logiciel *word2vec* (Mikolov *et al.*, 2013a), et il est disponible gratuitement en ligne [3]. Pour le manipuler, nous avons utilisé la librairie Python *Gensim* (Řehůřek & Sojka, 2010) [4]. L'autre raison pour laquelle nous avons travaillé sur l'anglais plutôt que le français est qu'il est morphologiquement moins riche, ce qui veut dire que le nombre d'occurrences pour chaque forme est plus élevé dans le corpus, et donc le nombre de contextes dans lesquels elle apparaît, augmentant ainsi la qualité des vecteurs (Bullinaria & Levy, 2007).

Nous avons ciblé trois formes verbales : la forme nue, la troisième personne du singulier du présent de l'indicatif, ainsi que le passé (*become*, *becomes*, *became*). Le gérondif (*–ing*) étant très ambigu, puisqu'il correspond souvent à la fois à

3. https://code.google.com/p/word2vec
4. https://radimrehurek.com/gensim

un nom et à un verbe, nous ne l'avons pas utilisé.

Un premier jeu de données nous servant à effectuer les tests a été construit à partir d'une liste des mots anglais les plus fréquents (Davies, 2010) [5]. De cette liste, 19 verbes qui n'avaient pas d'homonyme évident d'une autre partie du discours et un qui en avait (*live*, *lives*) ont été manuellement sélectionnés (cf. table 2).

Une autre liste de verbes a été compilée automatiquement à partir de 10 millions de mots tirés du *British National Corpus* (Burnard, 2007). L'ensemble des verbes du BNC ont été isolés et les variantes morphologiques ont été regroupées autour du lemme. La liste résultante comportait 782 verbes, dont 762 avaient leurs trois formes dans le vocabulaire du modèle de Mikolov *et al.* (2013b) ; ce sont ces 3×762 formes verbales qui constituent le jeu de données étendu.

Nous construisons d'abord à partir du modèle pré-entraîné une matrice où le vecteur de chaque mot-forme du vocabulaire occupe une rangée. Ensuite, pour chaque colonne de la table 2, nous construisons une sous-matrice avec seulement les vecteurs des mots-formes de cette colonne (qui ont tous la même flexion). Nous avons donc une matrice de 3 millions de vecteurs et trois sous-matrices de 20 vecteurs chacune.

Notre hypothèse est que les mots-formes représentés par les vecteurs d'une sous-matrice, puisqu'ils partagent tous la même flexion, doivent avoir certaines propriétés distributionnelles en commun qui les distinguent des autres mots-formes du vocabulaire. Ces propriétés doivent se refléter dans les valeurs des 300 dimensions des vecteurs. Quand on compare les vecteurs d'une sous-matrice flexionnellement homogène à ceux de la matrice complète, certaines de leurs dimensions doivent être relativement homogènes. On peut les identifier en cherchant des dimensions qui varient peu au sein des vecteurs de la sous-matrice comparativement à la matrice générale.

VB	**VBZ**	**VBD**
ask	asks	asked
be	is	was
become	becomes	became
begin	begins	began
bring	brings	brought
continue	continues	continued
follow	follows	followed
get	gets	got
give	gives	gave
have	has	had
hear	hears	heard
live	lives	lived
meet	meets	met
receive	receives	received
seem	seems	seemed
send	sends	sent
speak	speaks	spoke
tell	tells	told
understand	understands	understood
write	writes	wrote

TABLE 2 – Verbes choisis manuellement

Dans un premier temps, nous avons calculé, dans chaque sous-matrice, un « ratio flexionnel » pour chaque dimension. Ce ratio indique à quel point la dimension en question reflète le contenu flexionnel *vs* lexical du mot-forme. Plus une dimension est fortement associée à du contenu flexionnel, plus ce ratio est élevé. Il est calculé en comparant la variance de cette dimension dans la sous-matrice à sa variance dans la matrice de tout le modèle [6]. Si une dimension varie peu d'un vecteur à l'autre au sein de la sous-matrice alors qu'elle a une variance élevée dans le modèle en général, c'est un signe qu'elle est fortement liée à la flexion, qui est la propriété commune à tous les vecteurs de la sous-matrice. Le rapport V_m/V_s, où V_m est la variance de la dimension dans la matrice complète et V_s est la variance de cette même dimension dans la sous-matrice, sera plus élevé si la variance dans la sous-matrice est relativement plus faible que dans la matrice générale. Pour ramener ce rapport à des valeurs entre 0 et 1, nous utilisons la formule $\frac{tanh(V_m/V_s - k)+1}{2}$ pour calculer le ratio flexionnel de chaque colonne. La tangente hyperbolique ($tanh$) donne une courbe en S, et la constante k permet d'ajuster où l'on souhaite que le rapport V_m/V_s croise 0,5, c'est-à-dire à partir de quel rapport V_m/V_s on considère que la dimension est surtout associée à du contenu flexionnel. Nous avons testé plusieurs valeurs de k entre 0,1 et 10, et les résultats étaient systématiquement meilleurs plus on se rapprochait de 0. Nous n'avons pas testé de valeurs négatives parce que les ratios obtenus avec un k quasi-nul étaient déjà très près de 1.

On obtient alors, pour chaque sous-matrice, une liste de n ratios (pour un espace sémantique à n dimensions) entre 0 et 1 qui nous indiquent à quel point chaque dimension est associée à la flexion qui est commune aux mots-formes de la sous-matrice. Ensuite, nous calculons la moyenne par colonne de la sous-matrice afin d'obtenir le vecteur moyen de cet ensemble de mots-formes. Nous multiplions chaque dimension de ce vecteur moyen par les ratios obtenus précédemment, ce qui nous donne alors le vecteur flexionnel qui est commun à tous les vecteurs de la sous-matrice.

Pour les trois formes à l'étude, les ratios obtenus pour chaque dimension étaient très élevés. Pour k=1, nous avions des ratios entre 0,61 et 0,99997, avec une médiane de 0,88. Ces ratios diminuent quand on augmente k et augmentent quand on réduit cette constante et nos résultats étaient systématiquement meilleurs plus nous rapprochions k de 0. Comme nous

5. http://www.wordfrequency.info

6. Nous avons testé diverses mesures de la variance : variance, déviation standard, écart médian à la moyenne, et écart médian à la médiane. La variance donnait systématiquement de meilleurs résultats.

multiplions ces ratios par le vecteur moyen de la sous-matrice, il est apparu évident que nos résultats étaient meilleurs quand les vecteurs flexionnels se rapprochaient de la moyenne des vecteurs de la sous-matrice. Nous avons donc poursuivi en utilisant directement la moyenne d'une sous-matrice comme vecteur flexionnel. Cela présente l'avantage de ne pas nécessiter de calcul sur la matrice complète, qui est très lourde. Le vecteur moyen d'un groupe de vecteurs se trouve au « milieu » de ces vecteurs dans l'espace sémantique. En supposant que le contenu lexical des vecteurs les font dévier dans des directions indépendantes, leur milieu devrait correspondre à ce qu'ils ont en commun, c'est-à-dire leur flexion.

Nous avons comparé deux méthodes basées sur la moyenne pour calculer les vecteurs flexionnels. La première (nommée « A » dans les résultats ci-dessous) est la simple moyenne par colonne de la sous-matrice correspondant à une flexion donnée. La seconde méthode (« B » dans les résultats) est plus complexe. D'abord on calcule le vecteur moyen des trois formes fléchies pour chaque mot, ce qui nous donne une approximation du vecteur lexical. Ensuite, pour chaque forme fléchie, nous soustrayons de son vecteur l'approximation du vecteur lexical, ce qui nous donne une approximation de son vecteur flexionnel. Finalement, le vecteur correspondant à une flexion donnée est la moyenne des approximations flexionnelles obtenues à partir de toutes les formes qui portent cette même flexion.

Une fois que nous avons identifié nos trois vecteurs flexionnels (VB, VBZ et VBD) selon une des deux méthodes ci-dessus, nous les soustrayons de chaque vecteur dans la sous-matrice afin d'obtenir les vecteurs lexicaux. Comme nous avons trois formes fléchies pour chaque mot, en soustrayant de chaque vecteur initial les vecteurs flexionnels identifiés, nous obtenons trois vecteurs lexicaux pour chaque mot. On peut en faire la moyenne pour obtenir le vecteur lexical final. Au lieu de cela, nous nous servons de ces trois vecteurs pour évaluer la performance de notre méthode.

4 Évaluation

Nous avons utilisé deux méthodes pour évaluer notre travail. La première consiste à mesurer la différence entre la similarité des vecteurs des formes fléchies d'un même mot et celle entre les vecteurs lexicaux calculés à partir de ces formes. Pour chaque mot, nous avons trois vecteurs initiaux (avant traitement) et trois vecteurs lexicaux (après traitement). Nous calculons la moyenne des similarités cosinus entre chaque paire de vecteurs d'un même mot, avant et après traitement. En principe, si les vecteurs flexionnels que nous avons identifiés sont valides, alors en les soustrayant des vecteurs initiaux on devrait obtenir des vecteurs plus similaires qu'ils ne l'étaient au départ.

Afin de vérifier que ce n'est pas seulement parce qu'on soustrait des valeurs dans les vecteurs que nos vecteurs se rapproche après traitement, nous utilisons une quatrième sous-matrice (de 20 ou 762 verbes), celle-ci remplie de vecteurs choisis au hasard dans la matrice globale. Cette sous-matrice aléatoire est soumise au même traitement que les trois autres, mais comme il n'y a rien de commun aux vecteurs qui la composent, les « vecteurs flexionnels » identifiés ne correspondent à rien. La similarité cosinus moyenne entre ces vecteurs aléatoires et les trois vecteurs associés à chaque mot ne devrait donc pas augmenter significativement avant et après traitement.

La seconde méthode consiste à récupérer dans le modèle les plus proches voisins des vecteurs flexionnels que nous avons identifiés. On s'attend à ce que les voisins du vecteur VBD, par exemple, soient des verbes au passé, et que ceux de VBZ soient des verbes à la troisième personne du singulier.

5 Résultats et discussion

Les tables 3 et 4 ci-dessous donnent les résultats de nos expériences. Les différentes expériences sont identifiées par un code composé d'une lettre qui identifie l'approche utilisée (voir §3), suivie de deux nombres. Le premier indique le jeu de données qui a été utilisé pour identifier les vecteurs flexionnels (20 verbes sélectionnés manuellement ou 762 verbes tirés du corpus BNC). Le second indique le jeu de données dans lequel nous avons soustrait les vecteurs flexionnels pour identifier les vecteurs lexicaux. C'est dans ce jeu de données que nous avons mesuré la similarité des vecteurs avant et après traitement. Dans les deux tables, les trois dernières colonnes donnent les résultats du traitement dans la sous-matrice aléatoire qui sert de témoin dans nos expériences.

La méthode B-20-20 est celle qui donne les meilleurs résultats. On note une augmentation moyenne du score de similarité de 0,14, qui passe de 0,634 à 0,774, soit un gain moyen de 22% (autrement dit, la distance moyenne entre les vecteurs passe de 0,366 à 0,226, soit une réduction de 38%). L'augmentation du score de similarité atteint presque 14% pour B-762-20 et 12% pour A-20-20. Les deux méthodes testées fonctionnent donc bien, mais on obtient systématiquement de

Expérience	Avant	Après	Δ	$\text{Avant}_{\text{al.}}$	$\text{Après}_{\text{al.}}$	$\Delta_{\text{al.}}$
B-20-20	0,634	0,774	0,140	0,046	0,011	−0,035
B-762-20	0,634	0,720	0,086	0,043	0,003	−0,040
A-20-20	0,634	0,711	0,077	0,010	−0,007	−0,017
A-762-20	0,634	0,684	0,050	0,029	−0,015	−0,044
B-762-762	0,579	0,626	0,047	0,051	0,003	−0,048
A-20-762	0,579	0,591	0,013	0,044	−0,004	−0,048
A-762-762	0,579	0,587	0,009	0,052	0,003	−0,049
B-20-762	0,579	0,587	0,008	0,049	0,002	−0,047

TABLE 3 – Résultats de diverses méthodes (en ordre décroissant de performance)

meilleurs résultats quand on applique les vecteurs flexionnels identifiés pour isoler les vecteurs lexicaux dans les 20 verbes sélectionnés manuellement, peu importe la méthode ou le jeu de données utilisés pour identifier les vecteurs flexionnels. Contrairement aux verbes tirés du BNC, ceux de la liste courte ne sont pas ambigus (sauf LIVE) et la soustraction des vecteurs flexionnels verbaux est donc plus pertinente. Dans le cas de formes homographiques, les vecteurs peuvent contenir, pêle-mêle, de l'information flexionnelle verbale, nominale et/ou adjectivale. Une analyse qualitative des résultats obtenus avec les 762 verbes du BNC permet d'ailleurs de vérifier que la tête de la liste triée en ordre décroissant de gain de similarité contient moins de formes ambiguës que la fin de la liste.

En comparaison, l'application des deux méthodes sur la sous-matrice aléatoire donne toujours une diminution du score de similarité. Les augmentations que nous observons dans nos expériences sont donc bien liées à un apprentissage effectué sur des données homogènes.

La table 4 donne les résultats détaillés pour chacun des 20 verbes sélectionnés manuellement. Il est intéressant de noter que les formes de LIVE sont ambiguës et que ce sont elles qui obtiennent l'augmentation la plus négligeable de cette liste.

Mot-forme	Avant	Après	Δ	$\text{Avant}_{\text{al.}}$	$\text{Après}_{\text{al.}}$	$\Delta_{\text{al.}}$
be	0,523	0,715	0,192	0,033	0,005	−0,028
get	0,654	0,845	0,190	0,025	−0,008	−0,033
give	0,692	0,873	0,181	0,031	0,005	−0,026
bring	0,629	0,800	0,171	0,091	0,081	−0,010
have	0,620	0,787	0,167	0,061	0,037	−0,024
receive	0,682	0,841	0,159	0,077	0,053	−0,024
ask	0,684	0,843	0,158	0,001	−0,062	−0,063
begin	0,629	0,785	0,156	0,009	−0,026	−0,034
speak	0,637	0,793	0,155	0,062	0,012	−0,050
send	0,685	0,832	0,148	0,089	0,072	−0,017
hear	0,671	0,816	0,144	0,026	−0,032	−0,057
meet	0,637	0,773	0,136	0,054	0,022	−0,032
tell	0,613	0,745	0,132	0,087	0,023	−0,065
become	0,688	0,819	0,131	0,016	0,002	−0,015
continue	0,720	0,851	0,131	0,014	−0,021	−0,034
follow	0,582	0,709	0,127	0,125	0,108	−0,017
understand	0,616	0,718	0,101	0,141	0,105	−0,035
seem	0,688	0,788	0,099	0,058	0,048	−0,010
write	0,559	0,643	0,085	−0,020	−0,092	−0,072
live	0,471	0,507	0,036	−0,060	−0,106	−0,046
Moyenne	0,634	0,774	0,140	0,046	0,011	−0,035

TABLE 4 – Résultats détaillés de B-20-20 (en ordre décroissant de performance)

On note une corrélation modérée entre la similarité moyenne des formes d'un mot avant traitement et l'augmentation du score de similarité après traitement (les colonnes « Avant » et « Δ » dans les tables 3 et 4). Par exemple, pour B-762-762, cette corrélation est de 0,45. Ce n'est pas surprenant puisque la similarité moyenne des formes d'un mot a tendance à être

plus faible quand il y a une ou plusieurs formes ambiguës. Justement, ces formes ambiguës font obstacle à l'identification d'un vecteur flexionnel clair.

La table 5 donne les 20 plus proches voisins des vecteurs flexionnels VB, VBZ et VBD (en utilisant la méthode B sur 762 verbes). Ces voisins sont relativement distants (similarité cosinus entre 0,36 et 0,48). On voit clairement que le vecteur VB a été moins bien identifié que VBZ et VBD. C'est vraisemblablement dû au fait que la forme nue est elle-même ambiguë : il peut s'agir d'un infinitif, d'un présent de l'indicatif, d'un impératif, etc. On note que la plupart des voisins de VB sont des expressions à mots multiples. Nous croyons que cela s'explique par le fait que ces expressions sont peu fréquentes dans le corpus, et donc que leur vecteur est peu fiable. Autrement dit, notre vecteur VB se retrouve dans une zone mal définie de l'espace sémantique. Les deux autres formes sont moins ambiguës et donnent donc de meilleurs résultats. On peut ainsi supposer que notre méthode fonctionnerait mieux sur une langue morphologiquement plus riche, où les formes ont tendance à être moins ambiguës, à condition que le corpus soit assez gros pour assurer un nombre suffisant d'occurrences pour chaque forme.

VB	**VBZ**	**VBD**
OPTION_ONE	*sees*	*moved*
tofind	*creates*	*arrived*
Bedbugs_Bite_Act	*finds*	*turned*
through_emergency_recapitalizion	*introduces*	*returned*
outguess_God	*initiates*	*escorted*
contacting_Melissa_Medalie	*gets*	*shaken_awake*
Yourself_Loan_Modification	*uses*	*pushed*
Eat_Spaghetti_Dinner	*pushes*	*chased*
Tiger_Wear_Necktie	*sustains*	*hauled*
unleash_cyberattacks	*interprets*	*untouchable_Gio_Gonzalez*
Diet_Pepsi_Skinny	*embraces*	*exited*
cooperate_Ramuglia	*manages*	*snatched*
###-###-####_FAX_Afri	*amplifies*	*brought*
OK_Vanhaudenhuyse	*develops*	*stood*
overcomplicate_things	*takes*	*appeared*
Heartaches_Begin	*engages*	*had*
Them_Hear	*adopts*	*greeted_enthusiastically*
Er_Rip	*indulges*	*chief_Siegfried_Sievert*
injure_Pacioretty	*nurtures*	*approached*
react_robustly	*pulls*	*picked*

TABLE 5 – Vingt plus proches voisins des vecteurs flexionnels (B-762)

Il faut être prudent en comparant les résultats obtenus à partir des 20 verbes manuellement sélectionnés à ceux obtenus à partir du corpus BNC. En effet, il y a ici deux différences importantes : d'une part, le niveau d'ambiguïté, comme nous l'avons noté, mais aussi d'autre part la taille de l'échantillon. Pour vérifier que c'est bien l'ambiguïté qui est en jeu, on pourrait par exemple échantillonner 20 verbe parmi ceux du corpus BNC et répéter l'opération plus fois, puis faire la moyenne des résultats [7], ce que nous n'avons pas testé.

6 Conclusion

La décomposition d'un vecteur en un vecteur lexical et un vecteur flexionnel est une forme de « séparation aveugle de source » en traitement du signal. Il est probable que l'apprentissage machine soit utile pour cette tâche, mais nous avons proposé des techniques simples basées sur la moyenne des vecteurs. Notre approche donne de bons résultats quand on traite des vecteurs de formes non ambiguës, mais la performance diminue considérablement quand on a des formes homonymiques. Nous croyons que cela reflète la qualité des vecteurs initiaux et est lié aux limites inhérentes à la sémantique distributionnelle.

7. Nous remercions le relecteur anonyme qui a attiré notre attention sur ce point.

On peut se demander si les vecteurs flexionnels obtenus peuvent eux-mêmes être décomposés. Par exemple, il est possible que les vecteurs VBD et VBZ puissent être décomposés en un vecteur correspondant à la partie du discours et un autre correspondant au sens flexionnel.

Ici, nous avons travaillé à partir de vecteurs déjà entraînés sur un corpus non lemmatisé. Nous voulions ainsi éviter le bruit inévitablement introduit par un lemmatisateur. Il serait néanmoins utile de comparer nos résultats avec un modèle entraîné sur un corpus lemmatisé pour voir si les vecteurs ainsi obtenus se rapprochent de nos vecteurs lexicaux. Il serait également intéressant de tester notre approche sur des vecteurs construits à partir de corpus désambiguïsés.

Dans notre approche initiale à base de ratios, nous avons observé, pour les trois formes à l'étude, des ratios toujours très élevés. Cela indique que la distribution d'un mot est surtout conditionnée par sa flexion. Est-ce parce que le sens d'un mot-forme est surtout flexionnel, ou est-ce que les vecteurs contiennent en fait surtout de l'information morphosyntaxique ? Nous croyons que c'est plutôt la deuxième explication qui est la bonne, mais il y a matière à débat.

Références

BULLINARIA J. & LEVY J. (2007). Extracting semantic representations from word co-occurrence statistics : A computational study. *Behavior Research Methods*, **39**, 510–526.

L. BURNARD, Ed. (2007). *Reference Guide for the British National Corpus*. Research Technologies Service at Oxford University Computing Services.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, **12**, 2493–2537.

DAVIES M. (2010). Word frequency data : Corpus of contemporary american english.

DO Q.-K., ALLAUZEN A. & YVON F. (2014). Modèles de langue neuronaux : une comparaison de plusieurs stratégies d'apprentissage. In *Actes de la 21e conférence sur le traitement automatique des langues naturelles (TALN)*, p. 256–267, Marseille.

LAZARIDOU A., MARELLI M., ZAMPARELLI R. & BARONI M. (2013). Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, p. 1517–1526, Sophia.

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

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

MIKOLOV T., YIH W.-T. & ZWEIG G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 746–751, Atlanta, Georgia : Association for Computational Linguistics.

SCHWENK H. (2007). Continuous space language models. *Computer Speech & Language*, **21**(3), 492–518.

TURNEY P. D. & PANTEL P. (2010). From frequency to meaning : Vector space models of semantics. *Journal of Artificial Intelligence Research*, **37**(1), 141–188.

ŘEHŮŘEK R. & SOJKA P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, p. 45–50, Valletta, Malta : ELRA.