Université Paris Nanterre
École doctorale 139 – Connaissance, Langage, Modélisation

La résolution de la polysémie à l'aide de modèles de vecteur de mots et la visualisation de données : le cas des postpositions adverbiales *-ey*, *-eyse*, et *-(u)lo* en coréen

par Seongmin Mun

Thèse présentée et soutenue publiquement le 18 juin 2021
en vue de l'obtention du grade de
docteur en Traitement Automatique des Langues
sous la direction de Guillaume Desagulier

Membres du jury :

Directeur : Dr. Guillaume Desagulier	Université Paris VIII & UMR 7114, MoDyCo
Rapporteur : Prof. Iksoo Kwon	Hankuk University of Foreign Studies
Rapporteur : Prof. Laurent Prévot	Aix-Marseille Université
Examinatrice : Dr. Caroline Brun	Naver Labs Europe
Examinatrice : Prof. Iris Taravella	Université Paris Nanterre & UMR 7114, MoDyCo
Examinatrice : Prof. Delphine Battistelli	Université Paris Nanterre & UMR 7114, MoDyCo

Résumé

Ce projet de thèse présente des comptes rendus informatiques de la résolution de la polysémie au niveau des mots dans une langue peu étudiée—le Coréen. Les postpositions, qui se caractérisent par une correspondance forme-fonction multiple et qui sont donc polysémiques par nature, posent un défi à l'analyse automatique et à la performance des modèles pour identifier leurs fonctions. Dans ce projet, je consolide les modèles existants de classification de vecteur au niveau du mot (*Positive Pointwise Mutual Information* et *Singular Value Decomposition* ; *Skip-Gram and Negative Sampling*) en tenant compte du Window du contexte, et j'introduis un modèle de classification de vecteur au niveau de la phrase (*Bidirectional Encoder Representations from Transformers* (BERT)) dans le cadre de la modélisation sémantique distributionnelle. Par ailleurs, je développe deux systèmes de visualisation qui montrent (i) les relations entre les postpositions et leurs mots co-occurrents pour les modèles de vecteur au niveau du mot, et (ii) les clusters entre les phrases pour le modèle de vecteur au niveau de la phrase. Ces systèmes de visualisation ont l'avantage de mieux comprendre comment ces modèles de classification classent les fonctions prévues de ces postpositions. Les résultats montrent que, alors que la performance des modèles de vecteur au niveau du mot est modulée par la taille des corpus d'entraînement contenant les fonctions spécifiques des postpositions, le modèle de vecteur au niveau des phrases est stable (i.e., moins affecté par la taille du corpus) et simule la façon dont les humains reconnaissent la polysémie des postpositions adverbiales coréennes de façon plus appropriée que les modèles de vecteur au niveau du mot.

Mots-clés : polysémie, traitement automatique des langues, classification, modèles de vecteur de mots, visualisation de données, Coréen

1 Introduction

La polysémie, qui est un type d'ambiguïté, se produit lorsqu'une forme exprime des significations/fonctions multiples mais pourtant liées (Glynn and Robinson, 2014). Ce type de relation se retrouve également en coréen, une langue de la forme Sujet-Objet-Verbe avec une marquage explicite des cas par le biais d'une postposition dédiée (i.e., un morphème lié qui ajoute une signification grammaticale à un mot contenu où il est attaché ; Sohn, 1999). Une postposition coréenne implique normalement de nombreuses correspondances entre la forme et la fonction, et est de ce fait polysémique (Choo and Kwak, 2008). Par exemple, une postposition adverbiale *-ey*, l'une des postpositions étudiées dans cette thèse, est interprétée comme ayant huit fonctions majeures : localisation (LOC), but (GOL), effecteur (EFF), critère (CRT), thème (THM), instrument (INS), agent (AGT), et état final (FNS) (Shin, 2008). Supposons que la phrase suivante implique la postposition *-ey* comme fonction de LOC (localisation) tel que dans (1). Les locuteurs natifs du coréen (ou une personne ayant une bonne connaissance du coréen) peuvent facilement comprendre la fonction prévue de *-ey*.

(1) 지붕 위에 구멍이 났다.
cipung wi-ey kwumeng-i na-ss-ta.
Toit le dessus-LOC trou-NOM Il y a-PST-DECL
« Il y a un trou sur le dessus du toit. »

À cet égard, une question se pose quant à la manière dont un locuteur comprend la fonction de *-ey* en tant que LOC, compte tenu de ces diverses fonctions. Considérant que la signification d'un mot est étroitement liée à un contexte qui est créé par un groupe de mots voisins (DSMs ; Harris, 1954), cette question peut être résolue à travers le réseau de mots qui représente la relation entre les mots. Sur

la base de ce concept, plusieurs études ont cherché à saisir et à distinguer les différentes significations/fonctions des postpositions coréennes en appliquant des approches informatiques (e.g., Bae et al., 2014, 2015, Kim and Ock, 2016, Lee et al., 2015, Shin et al., 2005). Toutefois, les études antérieures ne se sont concentrées que sur l'amélioration des performances des modèles et n'ont pas essayé de comprendre comment ces modèles de classification classent les fonctions prévues de ces postpositions. Face à ce constat, je cherche, dans le cadre de cette thèse de doctorat, à appliquer des approches informatiques pour résoudre les problèmes que pose à la polysémie de ces postpositions au niveau du mot. De plus, afin de mieux comprendre comment ces modèles de classification classent les fonctions des postpositions, j'implémente des systèmes de visualisation qui montrent les vecteurs des mots et des phrases pour chaque modèle.

2 Contexte

2.1 Les modèles sémantiques distributionnels (DSMs)

L'idée fondamentale des DSMs repose sur le fait que la signification d'un mot est étroitement liée au contexte qui est créé par un groupe de mots voisins (Bullinaria and Levy, 2007, Turney and Pantel, 2010). Cette idée découle des premiers travaux en linguistique théorique de Harris (1954) et Firth (1957). Harris (1954) affirme que *les mots qui apparaissent dans des contextes similaires ont tendance à avoir des significations similaires*, tandis que Firth (1957) affirme que l'*on connaît un mot par son context*. Par exemple, *maison* et *appartement* apparaissent fréquemment avec des mots de contexte comme *loyer*, *chambre*, *vente*, etc., ce qui prouve aux modèles informatiques que *maison* et *appartement* peuvent être similaires l'un à l'autre. Il est important de noter que de nombreuses études ont signalé la force des modèles sémantiques distributionnels pour résoudre la polysémie du niveau des mots (e.g., Bae et al., 2015, Lee et al., 2015, Mun and Shin, 2020, Shin et al., 2005).

Dans son application réelle, les DSMs convertissent en vecteurs les informations contextuelles obtenues grâce aux mots situés autour d'un mot cible. Ils appliquent ensuite des algorithmes d'apprentissage automatique à ces vecteurs afin de mesurer la similarité sémantique de mot (e.g., Clark, 2015, Erk, 2012, Turney and Pantel, 2010). Les DSMs sont composés de deux types de modèles de vecteur de mot. Le premier est un modèle basé sur le comptage (e.g., Singular Value Decomposition (SVD) : Eckart and Young, 1936) qui est sensible à la fréquence des tokens (Jurafsky and Martin, 2019). Le second est un modèle basé sur la prédiction (e.g., Skip-Gram and Negative Sampling (SGNS) : Mikolov et al., 2013) qui s'appuie sur la fréquence des types (Mikolov et al., 2013). Les études antérieures ont montré que ces modèles de vecteur de mot ont l'avantage de représenter la relation entre les mots (e.g., Bae et al., 2015, Lee et al., 2015, Shin et al., 2005).

En plus de ces modèles traditionnels de vecteur de mot, des études récentes ont proposé un modèle de vecteur de mot contextualisé qui prend en compte les informations de voisinage d'un mot polysémique sur la base des séquences de mots autour du mot cible. Divers modèles ont été proposés pour cette tâche, tels que les *Embeddings de Language Models* (e.g., Peters et al., 2018), le *Generative Pre-Training* (e.g., Radford et al., 2018), et les *Bidirectional Encoder Representations of Transformer* (BERT ; Devlin et al., 2018). Parmi ces modèles, BERT montre les meilleures performances dans de nombreuses tâches telles que la traduction, la classification et la réponse à des questions (e.g., Devlin et al., 2018, Tang et al., 2019).

Sur la base de ces antécédents, j'utilise dans cette thèse une une combinaison de *Positive Pointwise Mutual Information* (PPMI ; Church and Hanks, 1989) et de la *Singular Value Decomposition* (SVD ; Eckart and Young, 1936) comme modèle basé sur le nombre, et du *Skip-Gram and Negative Sampling* (SGNS ; Mikolov et al., 2013) comme modèle basé sur la prédiction pour les modèles traditionnels de vecteur des mots. En outre, j'ai choisi BERT comme modèle de vecteur de motint *contextualisés* pour la tâche de

classification visant à identifier la fonction prévue d'une postposition dans une phrase.

2.2 La Polysémie dans les postpositions adverbiales coréennes

Le coréen, qui est la langue qui nous intéresse dans cette thèse, est une langue Sujet-Objet-Verbe avec le marquage explicite des cas par le biais d'une postposition—un morphème lié qui ajoute des fonctions grammaticales à un mot de contenu où il est attaché (Sohn, 1999). Les postpositions coréennes sont divisées en deux catégories : (i) grammaticale, indiquant les relations syntaxiques entre les mots du contenu et (ii) sémantique, indiquant les fonctions spécifiques selon le contexte de la phrase particulière (Sohn, 1999). Plus précisément, les postpositions adverbiales (classées comme postposition sémantique) sont polysémiques en raison de leur correspondance multiple entre forme et fonction, qui s'accompagne d'une ambiguïté fonctionnelle (Choo and Kwak, 2008). Parmi les diverses postpositions adverbiales, dans cette thèse, je me restreins à trois postpositions adverbiales : *-ey*, *-eyse* et *-(u)lo*, qui sont fréquemment utilisées en coréen et donc souvent documentées dans les études précédentes (e.g., Cho and Kim, 1996, Jeong, 2010, Nam, 1993, Park, 1999, Song, 2014).

2.2.1 *-ey*

Les études précédentes ont examiné les fonctions de *-ey* et ont proposé leurs propres affirmations concernant les types de fonctions impliquant *-ey*. Par exemple, Cho and Kim (1996) ont classé 10 types. Nam (1993) a affirmé que la relation entre un (pro-)nom et un prédicat combiné avec une postposition est importante pour déterminer sa fonction, ce qui a donné 14 types de postposition.

Afin de déterminer le nombre de fonctions de cette postposition, cette thèse met particulièrement l'accent sur huit fonctions majeures de *-ey* : localisation, but, effecteur, critère, thème, instrument, agent, et état final, qui sont fréquemment attestées dans le corpus Sejong, le corpus représentatif du Coréen.

2.2.2 *-eyse*

-eyse a moins de fonctions que les deux autres postpositions *-ey* et *-(u)lo* (Choo and Kwak, 2008). Cependant, la fréquence de son utilisation est également élevée par rapport à celle des autres (e.g., Cho and Kim, 1996, Song, 2014). Les chercheurs s'accordent généralement sur la fonction primaire comme étant le lieu qui s'engage dans le départ de l'action (e.g., Cho and Kim, 1996, Park et al., 2000, Song, 2014) et le corpus Sejong démontre également la même tendance. Les deux fonctions (source et lieu) sont majoritairement plus fréquentes que les autres.

2.2.3 *-(u)lo*

Les études antérieures ont examiné les fonctions de *-(u)lo* et ont proposé différents points de vue sur le nombre de fonctions pour *-(u)lo*. À titre d'exemple, Park (1999) affirme que la fonction centrale est instrumentale et qu'il existe neuf autres fonctions, telles que le chemin, la direction, le point de direction, le temps, le changement d'état, la qualification, le matériel, la cause et la manière. En revanche, Jeong (2010) place la fonction directionnelle au centre des différentes fonctions et explique la relation entre la fonction centrale et les fonctions étendues.

La classification du projet Sejong est quelque peu différente de ces deux études, puisqu'elle indique qu'il existe six fonctions majeures de *-(u)lo* : état final, instrument, direction, effecteur, critère, et localisation, avec les trois principales (état final ; instrument ; direction) occupant plus de 80% de l'utilisation totale. Car le corpus Sejong est largement utilisé dans les études sur le Coréen (e.g., Kang and Park, 2003, Kim et al., 2007, Park and Cha, 2017, Shin et al., 2005).

2.3 Les précédentes recherches en NLP sur les postpositions adverbiales

Les études sur la polysémie du niveau des mots en Coréen se sont principalement concentrées sur la catégorisation des différentes significations/fonctions des mots polysémiques pour l'interprétation essentielle des phénomènes linguistiques (e.g., Ahn, 1983, Hong, 1978, Lee, 1983, Maeng, 2016). Les chercheurs travaillant sur la linguistique informatique en Coréen suivent cette tendance et développent des systèmes qui classifient et reconnaissent automatiquement ces multiples significations/fonctions impliquant les mots afin de traiter les outils linguistiques d'une manière plus facile et plus efficace (e.g., Bae and Lee, 2015, Kang and Park, 2003, Kim and Ock, 2015, Lee et al., 2015, Shin et al., 2005). Par exemple, Lee et al. (2015) ont employé un SVM pour proposer un système d'étiquetage des rôles sémantiques. Dans cette étude, 4 096 phrases ont été utilisées pour l'apprentissage et 786 phrases ont été utilisées pour le test, ce qui a permis d'obtenir une précision de 0.77 pour la classification.

Malgré de nombreuses recherches sur la postposition adverbiale en Coréen, elles se sont surtout concentrées sur l'amélioration de la précision de la classification des fonctions et n'ont pas prêté attention à l'environnement des postpositions, comme les mots co-occurrents, qui génèrent un cluster centré autour de la postposition. D'un point de vue linguistique, une relation de groupes de mots liés entre eux est sans aucun doute une ressource linguistique précieuse car elle montre comment la polysémie est interprétée à travers eux. À cet égard, les modèles sémantiques distributionnels (DSMs ; Baroni et al., 2014), qui soutiennent que la signification d'un mot est étroitement liée à un contexte créé par un groupe de mots voisins, attirent l'attention sur la compréhension informatique dans le langage humain (Bullinaria and Levy, 2007, Turney and Pantel, 2010). Dans cette thèse, j'adopte l'idée que les DSMs fournissent des clusters entre le mot cible et les mots co-occurrents. Sur la base de ces DSM, j'améliore les approches précédentes de cette tâche en créant des modèles de classification basés sur des vecteur du niveau du mot—*Positive Pointwise Mutual Information* et *Singular Value Decomposition* (PPMI-SVD ; Turney and Pantel, 2010) et *Skip-Gram and Negative Sampling* (SGNS ; Mikolov et al., 2013)—ainsi que sur des vecteurs du niveau de la phrase—le *Bidirectional Encoder Representations of Transformers* (BERT ; Devlin et al., 2018). En outre, pour mieux comprendre comment ces modèles de classification reconnaissent les fonctions prévues des postpositions, ce projet met en œuvre des systèmes de visualisation qui montrent les relations des mots et des phrases pour chaque modèle.

3 Mise en place méthodologique : PPMI-SVD et SGNS

3.1 Corpus

3.1.1 Création du corpus annoté

Dans cette thèse, j'utilise le corpus de données représentatif du Coréen connu sous le nom de corpus Sejong (Kim et al., 2006). Cependant, le corpus Sejong ne code pas directement l'information sur les fonctions des postpositions dans chaque phrase (ce qui est nécessaire pour l'entraînement du modèle). Par conséquent, j'annote le corpus manuellement avec l'aide de trois locuteurs natifs du Coréen. Parmi les trois, l'un était un professeur qui enseigne le coréen aux enfants et les deux autres étaient des candidats au doctorat en linguistique. Ils ont géré tous les détails de l'annotation du corpus, depuis le développement du manuel d'annotation jusqu'à l'annotation manuelle de la fonction prévue de la postposition dans chaque phrase.

En ce qui concerne le processus de création d'un corpus codé à la main, j'extrais les phrases n'ayant qu'une seule postposition et un seul prédicat. Bien que cette manipulation ait permis d'omettre de nombreuses phrases déjà extraites du corpus original, elle a été bénéfique pour contrôler tout facteur de

confusion supplémentaire qui aurait pu interférer avec les performances de mon modèle. Si une phrase contient plus d'une postposition, y compris les trois postpositions sur lesquelles je me suis attardé, elles deviennent moins indépendantes les unes des autres. Cela signifie que les performances du modèle de chaque postposition seront affectées les unes par les autres. Ce processus de réduction a donné lieu à un total de 27,720 phrases, dont 14,096 phrases pour *-ey*, 5,078 phrases pour *-eyse* et 8,546 phrases pour *-(u)lo*. J'extrais ensuite 5,000 phrases au hasard pour chaque postposition afin de conserver un nombre égal de phrases pour chacune d'entre elles.

Tableau 1 – Liste de fréquence des sous-fonctions de *-ey*, *-eyse*, et *-(u)lo* dans le corpus validé par croisement

-ey		*-eyse*		*-(u)lo*	
Fonction	Fréquence	Fonction	Fréquence	Fonction	Fréquence
LOC	1,780	LOC	4,206	FNS	1,681
CRT	1,516	SRC	647	DIR	1,449
THM	448			INS	739
GOL	441			CRT	593
FNS	216			LOC	158
EFF	198			EFF	88
INS	69				
AGT	47				
Total	4,715	Total	4,853	Total	4,708

Note. Abréviation : AGT= agent ; CRT= critère ; DIR= direction ; EFF= effecteur ; FNS= état final ; GOL= but ; INS= instrument ; LOC= localisation ; SRC= source ; THM= thème

Les données du corpus final sont ensuite codées à la main par les trois locuteurs natifs du coréen, en suivant les fonctions des différentes postpositions. La fiabilité inter-juges des données a été mesurée avec le Fleiss's Kappa (Landis and Koch, 1977). Les résultats sont un score de 0.948 pour *-ey*, 0.928 pour *-eyse*, et 0.947 pour *-(u)lo*, qui sont considérés comme '*presque parfaits*' selon l'échelle de Kappa. Je décide ensuite d'exclure les phrases qui ont provoqué un désaccord entre les annotateurs humains (c'est-à-dire 285 phrases pour *-ey*, 147 phrases pour *-eyse*, et 292 phrases pour *-(u)lo*). Après lequel, j'obtiens les données du corpus final pour chaque postposition. Cela a donné 4,715 phrases pour *-ey*, 4,853 phrases pour *-eyse*, et 4,708 phrases pour *-(u)lo*. Le Tableau 1 présente la liste détaillée des fréquences par fonction des trois types de postpositions [1].

3.1.2 Création des ensembles de formation et de test

Chaque instance du corpus annoté a été lemmatisée et marquée par l'étiquetage morpho-syntaxiquement (aussi appelé étiquetage grammatical, POS tagging (part-of-speech tagging) en anglais) avant l'étape de traitement des données proprement dite. L'utilisation du corpus pour cette tâche exige que les fonctions de chaque postposition soient marquées ouvertement avec la forme de chaque postposition (e.g., *에/JKB_CRT*). Par conséquent, je marque les fonctions des postpositions manuellement.

Les données pour la formation et le test doivent être indépendantes les unes des autres. Ainsi, je divise le corpus en deux sous-ensembles, l'un avec 90% du corpus pour la formation et les 10% restants pour les tests. Afin d'obtenir un résultat normalisé de chaque modèle, j'utilise la technique de *validation croisée à k blocs* (Salton, 1971), qui évalue le modèle en partitionnant le corpus original en *k* sous-

1. Le corpus codé à la main est disponible à l'adresse suivante : https://github.com/seongmin-mun/Corpora/tree/master/APIK

échantillons de taille égale. Je fixe la valeur de *k* à 10 et je répète la validation croisée 10 fois, avec chacun des 10 sous-échantillons utilisés exactement une fois comme le jeu de test.

3.2 Formation du modèle

Pour le modèle de classification, j'ai employé PPMI-SVD (Turney and Pantel, 2010) et SGNS (Mikolov et al., 2013) sur la base de l'estimation basée sur la similarité (Dagan et al., 1995) pour le dressage du modèle, en suivant un modèle sémantique distributionnel (DSM; Baroni et al., 2014). L'entraînement du modèle se compose de deux parties : (i) vecteur au niveau des mots pour vérifier la relation entre les mots, et (ii) l'estimation basée sur la similarité (Dagan et al., 1995) pour déterminer les fonctions prévues de la postposition utilisée dans l'ensemble de test.

3.2.1 Des vecteurs au niveau des mots : PPMI-SVD et SGNS

Le flux général pour des vecteurs au niveau du mot est le suivant. Tout d'abord, le modèle crée une liste de mots qui existent dans les ensembles de formation obtenus par la technique de validation croisée à 10 blocs. Deuxièmement, sur la base de la liste de mots, une matrice de co-occurrence mot-mot (pour le modèle basé sur le comptage) et des vecteurs à un coup (pour le modèle basé sur la prédiction) sont générés. Troisièmement, le modèle produit des vecteurs au niveau du mot en utilisant PPMI-SVD et SGNS.

Le premier algorithme des vecteurs au niveau du mot a été développé dans un environnement Python. *Linalg* du package *scipy* a été utilisé pour la formation du modèle PPMI-SVD. *Word2Vec*, du package *gensim*, a été utilisé pour la formation du modèle SGNS. Les vecteurs au niveau des mots générés par chaque modèle avaient 500 dimensions, chacune d'entre elles étant stockée dans une base de données. Un total de 600 vecteurs a été réalisé par cet algorithme (2 modèles * 3 postpositions * 10 plis * 10 tailles de Window).

3.2.2 Estimation basée sur la similarité

Sur la base des vecteurs au niveau du mot généré par le premier algorithme, le second algorithme a été développé pour classifier la fonction prévue des postpositions utilisées dans l'ensemble de test. Ceci a été fait en calculant l'estimation basée sur la similarité (Dagan et al., 1995) : classer le sens du mot cible qui n'a jamais été utilisé dans les ensembles d'entraînement en utilisant les scores de similarité calculés entre les mots. Dans cette thèse, j'ai utilisé la formule de similarité en cosinus pour calculer le score de similarité entre une postposition et ses mots co-occurrents.

L'algorithme pour l'estimation basée sur la similarité se déroule comme suit. Tout d'abord, l'algorithme charge un total de 600 vecteurs au niveau du mot (2 modèles * 3 postpositions * 10 plis * 10 tailles de Window) générés par le premier algorithme et calcule la similarité entre les postpositions et les mots environnants. Deuxièmement, l'algorithme charge un jeu de test et établit la liste des mots qu'il contient. Troisièmement, l'algorithme compare la liste de mots utilisée dans l'ensemble de test à celle utilisée dans l'ensemble d'apprentissage et génère une liste de mots qui sont partagés entre eux. Quatrièmement, l'algorithme calcule le score moyen entre chaque fonction des postpositions et une liste de mots qui sont partagés entre eux. Enfin, l'algorithme détermine la fonction des postpositions utilisée dans le jeu de test avec la moyenne la plus élevée [2].

2. Le code complet des modèles des vecteurs au niveau du mot que j'ai développés sont disponible sur le site : https://github.com/seongmin-mun/PhD_dissertation/tree/main/Python/PPMI-SVD et https://github.com/seongmin-mun/PhD_-dissertation/tree/main/Python/SGNS

3.3 Visualisation : PostEmbedding

Afin d'interpréter intuitivement les clusters entre les postpositions et les mots qui les entourent, j'ai développé un système de visualisation (disponible sur le site : PostEmbedding). Dans le but d'exprimer les vecteurs au niveau du mot des DSM dans la visualisation bidimensionnelle, j'ai utilisé le *t-distributed Stochastic Neighbor Embedding* (t-SNE ; Maaten and Hinton, 2008) pour la réduction de la dimension des vecteurs au niveau du mot. Ces résultats ont été introduits dans le système de visualisation. Le système a été développé à l'aide des environnements JavaScript, HTML et CSS [3].

4 Résultats : les vecteurs au niveau du mot

4.1 Performance du modèle : Classification

4.1.1 PPMI-SVD

Les Tableaux suivants (Tableaux 2-4) montrent la précision de classification du modèle PPMI-SVD pour chaque postposition. Les résultats montrent que le modèle est plus performant pour *-eyse* que pour les deux autres postpositions (*-ey* et *-(u)lo*). La précision moyenne de classification pour *-ey*, *-eyse* et *-(u)lo* est d'environ 0.534, 0.773 et 0.567 respectivement.

Tableau 2 – Précision par fonction pour le modèle PPMI-SVD : *-ey*

taille du Window	Précision de classification								
	Overall	*AGT*	*CRT*	*EFF*	*FNS*	*GOL*	*INS*	*LOC*	*THM*
1	0.470	0.675	0.551	0.453	0.438	0.555	0.317	0.396	0.430
3	0.432	0.625	0.438	0.558	0.448	0.609	0.317	0.380	0.377
5	0.554	0.575	0.585	0.584	0.329	0.561	0.250	0.607	0.359
7	0.597	0.575	0.588	0.537	0.267	0.489	0.183	0.765	0.298
9	0.600	0.475	0.576	0.532	0.257	0.491	0.167	0.780	0.323
10	0.600	0.475	0.580	0.532	0.267	0.493	0.183	0.776	0.318
Moyenne	0.534	0.578	0.544	0.541	0.337	0.538	0.238	0.602	0.345

Tableau 3 – Précision par fonction pour le modèle PPMI-SVD : *-eyse*

taille du Window	Précision de classification		
	Overall	*LOC*	*SRC*
1	0.634	0.627	0.678
3	0.648	0.643	0.680
5	0.809	0.876	0.367
7	0.859	0.970	0.130
9	0.859	0.980	0.062
10	0.856	0.977	0.062
Moyenne	0.773	0.838	0.353

3. Plus de détails sur le PostEmbedding sont disponibles sur le site : https://github.com/seongmin-mun/VisualSystem/tree/master/Major/PostEmbedding

Tableau 4 – Précision par fonction pour le modèle PPMI-SVD : *-(u)lo*

taille du Window	Précision de classification						
	Overall	*CRT*	*DIR*	*EFF*	*FNS*	*INS*	*LOC*
1	0.480	0.497	0.547	0.462	0.425	0.491	0.353
3	0.532	0.552	0.730	0.562	0.394	0.482	0.347
5	0.572	0.426	0.840	0.438	0.530	0.348	0.220
7	0.608	0.313	0.855	0.312	0.714	0.248	0.140
9	0.608	0.262	0.833	0.238	0.767	0.221	0.140
10	0.607	0.257	0.830	0.238	0.770	0.220	0.127
Moyenne	0.567	0.405	0.777	0.391	0.583	0.344	0.233

L'analyse statistique des comparaisons en couple (Tableau 5) a également montré que la performance en *-eyse* était significativement meilleure que celle des deux autres postpositions. En revanche, l'exactitude de *-ey* et *-(u)lo* étaient statistiquement les mêmes.

Tableau 5 – Comparaison statistique de chaque postposition (PPMI-SVD) : t-test à deux échantillons

Comparaison	$\|t\|$	p
-ey vs. *-eyse*	6.080	$<$ .001***
-ey vs. *-(u)lo*	1.208	.243
-eyse vs. *-(u)lo*	5.929	$<$ .001***

Note. *** $<$ .001

Comme le montrent les Tableaux (Tableaux 2-4), pour la relation entre la taille du Window de contexte et les performances du modèle PPMI-SVD, j'ai constaté que la précision du modèle PPMI-SVD augmentait avec la taille du Window de contexte.

En outre, j'ai constaté que la performance des modèles de fonctions pour chaque postposition variait. La précision moyenne de classification de chaque fonction pour *-ey* est la plus élevée en LOC (0.602) et la plus faible en INS (0.238) ; pour *-eyse*, elle est la plus élevée en LOC (0.838) et la plus faible en SRC (0.353) ; pour *-(u)lo*, elle est la plus élevée en DIR (0.777) et la plus faible en LOC (0.233) (Tableaux 2-4). Si l'on considère que LOC pour *-ey*, LOC pour *-eyse* et DIR pour *-(u)lo* représentent la plus grande partie du corpus entier que les autres fonctions (voir Tableau 1), on peut indiquer que la performance du modèle PPMI-SVD a été affectée par la taille du corpus de chaque fonction.

4.1.2 SGNS

Comme pour le modèle PPMI-SVD, *-eyse* a surclassé les deux autres postpositions dans le modèle SGNS, comme le montrent les Tableaux (Tableaux 6-8). Cela s'est produit pour la même raison que pour le modèle PPMI-SVD (i.e., *-eyse* n'a que deux fonctions avec LOC qui occupent la majorité de la taille totale du corpus). La précision moyenne de classification pour *-ey*, *-eyse* et *-(u)lo* est d'environ 0.204, 0.693 et 0.368 respectivement.

Tableau 6 – Précision par fonction pour le modèle SGNS : *-ey*

taille du Window	Précision de classification								
	Overall	*AGT*	*CRT*	*EFF*	*FNS*	*GOL*	*INS*	*LOC*	*THM*
1	0.170	0.675	0.052	0.458	0.167	0.625	0.417	0.132	0.073
3	0.224	0.925	0.145	0.558	0.162	0.693	0.117	0.166	0.102
5	0.220	0.925	0.095	0.626	0.233	0.639	0.150	0.196	0.093
7	0.201	0.875	0.079	0.637	0.238	0.577	0.150	0.175	0.089
9	0.188	0.825	0.071	0.632	0.290	0.564	0.150	0.148	0.086
10	0.187	0.900	0.059	0.584	0.281	0.564	0.183	0.154	0.095
Moyenne	0.204	0.878	0.089	0.593	0.224	0.616	0.183	0.169	0.092

Tableau 7 – Précision par fonction pour le modèle SGNS : *-eyse*

taille du Window	Précision de classification		
	Overall	*LOC*	*SRC*
1	0.244	0.131	0.988
3	0.612	0.578	0.834
5	0.735	0.736	0.727
7	0.829	0.881	0.491
9	0.849	0.918	0.394
10	0.851	0.919	0.406
Moyenne	0.693	0.699	0.655

Tableau 8 – Précision par fonction pour le modèle SGNS : *-(u)lo*

taille du Window	Précision de classification						
	Overall	*CRT*	*DIR*	*EFF*	*FNS*	*INS*	*LOC*
1	0.345	0.507	0.769	0.550	0.011	0.147	0.247
3	0.396	0.554	0.874	0.538	0.039	0.154	0.313
5	0.374	0.468	0.797	0.662	0.072	0.124	0.420
7	0.362	0.455	0.739	0.700	0.087	0.121	0.480
9	0.348	0.461	0.688	0.725	0.076	0.132	0.553
10	0.349	0.492	0.695	0.725	0.060	0.137	0.540
Moyenne	0.368	0.499	0.774	0.634	0.058	0.141	0.427

Pourtant, contrairement aux résultats du modèle PPMI-SVD, l'analyse statistique des comparaisons en couple (Tableau 9) montre que les niveaux d'exactitude de tous les postpositions étaient différents.

Tableau 9 – Comparaison statistique de chaque postposition (SGNS) : t-test à deux échantillons

Comparaison	$\|t\|$	p
-ey vs. *-eyse*	7.835	$<$ .001***
-ey vs. *-(u)lo*	18.74	$<$ .001***
-eyse vs. *-(u)lo*	5.203	$<$ .001***

Note. *** $<$ .001

En ce qui concerne le rôle de la taille du Window contextuelle dans le modèle SGNS, contrairement aux

résultats du modèle PPMI-SVD, le taux de changement de la précision par postpositions semble stable, sauf pour *-eyse*, sans changement significatif lorsque la taille du Window augmente. Toutefois, comme le montre le Tableau 7, pour *-eyse*, le modèle SGNS présente une tendance similaire à celle du modèle PPMI-SVD, à savoir que la précision de chaque modèle augmente avec la taille du Window contextuelle.

De plus, comme pour le modèle PPMI-SVD, la performance du modèle SGNS varie également en fonction des fonctions de chaque postposition. La précision de classification moyenne de chaque fonction pour *-ey* est la plus élevée pour AGT (0.878) et la plus faible pour CRT (0.089) ; pour *-eyse*, elle est la plus élevée pour LOC (0.699) et la plus faible pour SRC (0.655) ; pour *-(u)lo*, elle est la plus élevée pour DIR (0.774) et la plus faible pour FNS (0.058) (Tableaux 6-8). Comme dans le cas du modèle PPMI-SVD, on peut également constater que la performance du modèle SGNS est affectée par la taille du corpus de chaque fonction.

4.2 Système de visualisation : clusters et mots co-occurrents

Le système de visualisation visait à identifier des vecteurs au niveau du mot de manière interactive afin de voir les changements des clusters entre chaque fonction des postpositions et les mots co-significatifs. Pour explorer statistiquement les changements des clusters par modèle et par taille de Window, j'ai effectué une série d'analyses de cluster en utilisant le clustering basé sur la densité (Sander et al., 1998).

Les résultats de la visualisation ont montré que les mots les plus fréquemment utilisés dans le corpus ont été placés au centre du cluster pour le modèle PPMI-SVD. C'est parce qu'il fonctionnait sur la base de la fréquence des jetons. En revanche, le modèle SGNS était basé sur la fréquence de type, et les mots ont été distribués sur toutes les tailles de Window, indépendamment de la fréquence des jetons. Toutefois, l'analyse des grappes a montré que les cartes sémantiques de distribution pour chaque modèle n'étaient pas tellement différentes en ce qui concerne le produit final du regroupement (produisant un ou deux groupes pour chaque modèle, ce qui indique que les grappes créées ne différaient pas significativement les unes des autres par environnement.

En outre, grâce à cette visualisation, j'ai découvert qu'il existait deux types de relations différentes entre la postposition particulière et ses mots co-occurrents : (i) les mots présentant une forte similarité mais une faible fréquence de co-occurrence, et (ii) les mots présentant une forte similarité et également une fréquence de co-occurrence élevée.

4.3 Question des modèles du vecteur au niveau du mot

Malgré ces résultats, les deux modèles que j'ai testés dans cette section présentent de sérieuses limitations. La performance des modèles est insatisfaisante au regard des études précédentes sur la classification des postpositions sur lesquelles je me suis concentré (e.g., Bae et al., 2015, Kim and Ock, 2016, Shin et al., 2005). Ils ont rapporté un niveau de précision allant de 0.882 (Kang and Park, 2003) à 0.623 (Bae et al., 2014). En revanche, le niveau moyen pour mes modèles était de 0.550. En outre, le modèle semble bien fonctionner uniquement lorsque les fonctions cibles apparaissent très fréquemment dans les données, ce qui n'est pas la façon dont je visais à traiter la résolution de la polysémie. Cela est dû à la nature technique de le vecteur du niveau du mot, qui distingue les mots présents dans l'ensemble du corpus en utilisant uniquement les informations morphologiques et la taille du Window, et qui utilise les mots sans tenir compte de leurs éventuels effets différents sur la détermination de la signification d'une postposition particulière. En effet, les modèles du vecteur au niveau du mot traditionnels sont *statiques*—un vecteur unique est attribué à chaque mot (e.g., Ethayarajh, 2019, Liu et al., 2019a).

Pour surmonter ces problèmes, j'ai employé BERT (Devlin et al., 2018) pour la classification des fonctions des postpositions. BERT produit des vecteurs contextuels, et cette caractéristique peut nous ai-

der à créer un meilleur système de classification des postpositions. Une tendance récente pour traiter cette tâche est appelée vecteur contextuel du mot, qui convertit tous les mots dans chaque vecteur en considérant le contexte (e.g., la position, une forme du mot) dans lequel ils apparaissent. Divers modèles ont été proposés, tels que *Embeddings from Language Models* (ELMo ; Peters et al., 2018) et *Generative Pre-Training* (GPT ; Radford et al., 2018), mais BERT montre la meilleure performance parmi tous les modèles présentés jusqu'à présent. Par conséquent, j'ai appliqué BERT à mon modèle de classification afin d'améliorer les performances du modèle.

5 Mise en place méthodologique : BERT

5.1 Corpus

J'ai prétraité les données en tenant compte du fonctionnement du BERT (j'ai utilisé le modèle original du BERT pour cette tâche). Tout d'abord, j'ai ajouté [CLS] ('classification' ; indiquant le début d'une phrase) avant une phrase et [SEP] ('séparation' ; indiquant la fin d'une phrase) après une phrase pour indiquer où la phrase commence et se termine. Ces indicateurs ont permis au modèle BERT de reconnaître une limite de phrase dans un texte, permettant au modèle d'apprendre le sens des mots en tenant compte des variations inter-sententielles. Ensuite, j'ai créé une colonne séparée ('Label') pour indiquer la fonction prévue de chaque postposition dans chaque phrase. J'ai par la suite divisé le corpus en deux sous-ensembles, l'un avec 90% du corpus pour l'entraînement et l'autre avec les 10% restants pour les tests.

5.2 Formation du modèle

J'ai défini les paramètres liés à l'entraînement de BERT tels que *batch size* (32), *epoch* (50), *seed* (42), *epsilon* (0.00000008), et *learning rate* (0.00002), comme conseillé par McCormick (2019). J'ai ensuite employé un modèle linguistique pré-entraîné afin d'obtenir une grande précision des résultats ; à cette fin, j'ai utilisé un modèle BERT coréen (KoBERT ; Jeon et al., 2019).

Ensuite, l'entraînement du modèle s'est déroulé comme suit. Tout d'abord, j'ai chargé KoBERT par le biais de la fonction *BertForSequenceClassification* de *Transformers* (Wolf et al., 2019). Deuxièmement, j'ai affiné le modèle pré-entraîné en utilisant l'ensemble d'entraînement, en vue de réduire les valeurs de perte et de mettre à jour le taux d'apprentissage pour une meilleure précision de classification du modèle. Troisièmement, j'ai chargé l'ensemble de test pour évaluer si le modèle affiné a reconnu avec succès les fonctions prévues de chaque postposition dans chaque phrase. Dans cette partie, les taux de précision pour chaque fonction et le taux de précision total ont été calculés en comparant la fonction prévue de chaque postposition dans chaque phrase test avec la fonction classée de chaque postposition via le modèle BERT. Enfin, j'ai employé *t-SNE* pour la réduction de la dimension des vecteurs de classification de la postposition par chaque *epoch*. En outre, pour confirmer statistiquement les changements des résultats de vecteur du niveau des phrases pour chaque *epoch*, j'ai effectué un *density-based clustering*. Ces résultats ont été introduits dans le système de visualisation [4].

5.3 Visualisation : PostBERT

Afin de voir comment le BERT comprend la polysémie du niveau des mots de chaque postposition, j'ai conçu un système de visualisation avec des environnements JavaScript, HTML et CSS, en utilisant l'en-

4. Le code complet de la formation du BERT que j'ai développé est disponible sur le site : https://github.com/seongmin-mun/PhD_dissertation/tree/main/Python/BERT

semble de test sous la distribution bidimensionnelle (disponible sur le site : PostBERT) [5]. Pour l'interface du système, j'ai créé trois zones pour la démonstration des performances du modèle : une carte de distribution pour les vecteurs au niveau de la phrase, des graphiques de précision/perte relatifs au modèle, et des graphiques pour le *density-based clustering*.

6 Résultats : Vecteur au niveau de la phrase

6.1 Performance du modèle : Classification

Les tableaux suivants (Tableaux 10-12) montrent la précision de classification du modèle BERT pour chaque postposition. Les résultats montrent que le modèle BERT a mieux fonctionné pour *-eyse*, qui n'a que deux fonctions (SRC et LOC), que pour les deux autres postpositions (*-ey* et *-(u)lo*). La précision moyenne de classification pour *-ey*, *-eyse* et *-(u)lo* est d'environ 0.815, 0.898 et 0.813 respectivement. Il s'agit d'un niveau de précision satisfaisant si l'on considère que les études précédentes sur la classification des postpositions ont rapporté un niveau de précision allant de 0.621 (Bae et al., 2014) à 0.837 (Kim and Ock, 2016).

Tableau 10 – Précision par fonction pour le modèle BERT : *-ey*

Epoch	Précision de classification								
	Overall	*AGT*	*CRT*	*EFF*	*FNS*	*GOL*	*INS*	*LOC*	*THM*
1	0.682	0	0.876	0	0	0.044	0	0.911	0.198
10	0.819	0	0.930	0.433	0.578	0.313	0.133	0.954	0.688
20	0.817	0.067	0.897	0.533	0.533	0.186	0.067	0.960	0.916
30	0.824	0.067	0.915	0.378	0.444	0.328	0.067	0.948	0.718
40	0.826	0.067	0.892	0.489	0.467	0.326	0.133	0.942	0.768
50	0.824	0.067	0.912	0.411	0.389	0.409	0.1	0.940	0.683
Moyenne	0.815	0.041	0.911	0.439	0.497	0.328	0.076	0.947	0.713

Tableau 11 – Précision par fonction pour le modèle BERT : *-eyse*

Epoch	Précision de classification		
	Overall	*LOC*	*SRC*
1	0.863	0.980	0.174
10	0.9	0.939	0.559
20	0.898	0.937	0.651
30	0.896	0.949	0.464
40	0.912	0.963	0.523
50	0.916	0.960	0.598
Moyenne	0.898	0.948	0.535

5. Plus de détails sur le PostEmbedding sont disponibles sur le site : https://github.com/seongmin-mun/VisualSystem/tree/master/Major/PostBERT

Tableau 12 – Précision par fonction pour le modèle BERT : *-(u)lo*

Epoch	Précision de classification						
	Overall	*CRT*	*DIR*	*EFF*	*FNS*	*INS*	*LOC*
1	0.704	0.476	0.943	0	0.764	0.477	0
10	0.814	0.83	0.918	0.367	0.771	0.835	0.1
20	0.812	0.694	0.951	0.3	0.838	0.709	0.044
30	0.816	0.708	0.941	0.333	0.811	0.752	0.05
40	0.819	0.694	0.927	0.267	0.855	0.777	0.05
50	0.821	0.692	0.957	0.4	0.836	0.723	0.1
Moyenne	0.813	0.721	0.938	0.278	0.815	0.763	0.106

Pour explorer statistiquement la classification par postpositions/epochs, j'ai effectué un *t*-test à deux échantillons. Comme le montre le Tableau 13, la performance du modèle pour *-eyse* est significativement meilleure que pour les deux autres postpositions. Compte tenu du nombre différent de fonctions (e.g., deux pour *-eyse*, six pour *-(u)lo*, et huit pour *-ey*), ce résultat indique une relation inverse entre la précision de la classification et le nombre de fonctions que chaque postposition manifeste.

Tableau 13 – Comparaison statistique de chaque postposition (BERT) : t-test à deux échantillons

Comparaison	$\lvert t \rvert$	p
-ey vs. *-eyse*	22.588	$<$ .001***
-ey vs. *-(u)lo*	0.533	$<$ .594
-eyse vs. *-(u)lo*	28.301	$<$ .001***

Note. *** $<$ .001

De plus, la précision moyenne de classification de chaque fonction pour *-ey* est la plus élevée pour LOC (0.947) et la plus faible pour AGT (0.041) ; pour *-eyse*, elle est la plus élevée pour LOC (0.948) et la plus faible pour SRC (0.535) ; pour *-(u)lo*, elle est la plus élevée pour DIR (0.938) et la plus faible pour LOC (0.106) (Tableaux 10-12). Quant aux occurrences des fonctions individuelles par postposition, LOC pour *-ey*, LOC pour *-eyse*, et DIR pour *-(u)lo* représentent la plus grande partie du corpus entier que les autres fonctions (voir Tableau 1). Ce résultat indique donc que la performance du modèle a été affectée par les proportions asymétriques des fonctions composant l'utilisation de chaque postposition.

6.2 Système de visualisation : clusters du vecteur au niveau de la phrase

Le système de visualisation a montré que le modèle était capable de reconnaître les fonctions de chaque postposition au fur et à mesure de la progression de l'epoch. Pour *-ey*, toutes les phrases étaient divisées en deux groupes lorsque l'epoch était la première, mais au fur et à mesure que l'epoch progressait, les phrases étaient divisées en trois à l'epoch 7, quatre à l'epoch 12 et cinq à l'epoch 15. Pour *-eyse*, le nombre de groupes était de un lorsque l'epoch était la première, et il y avait deux groupes lorsque l'epoch était la neuvième. Pour *-(u)lo*, le nombre de clusters a augmenté, passant de un (Epoch 1) à trois (Epoch 4), cinq (Epoch 12), et six (Epoch 46).

En particulier, pour *-(u)lo*, j'ai fait deux découvertes intéressantes. Tout d'abord, à l'epoch 12, un groupe de fonctions EFF (fonctions dont les occurrences sont peu fréquentes dans les données) est apparu. Ce résultat indique que l'ORET peut identifier des fonctions à un niveau satisfaisant, même si elles sont relativement peu fréquentes, à condition que le nombre d'epochs fournies soit suffisant.

Deuxièmement, il est intéressant de noter que LOC n'a pas pu former un groupe désigné au final. En mettant en évidence et en zoomant sur les instances individuelles de LOC, j'ai constaté que de nombreuses instances de LOC (11 sur 15) appartenaient au groupe DIR. Cela est dû à (i) la faible fréquence des LOC dans les données et (ii) la proximité sémantique entre DIR et LOC—ils se rapportent à un lieu et sont souvent difficiles à distinguer les uns des autres. Ce résultat indique que l'identification des fonctions est encore limitée par les complications mentionnées ci-dessus.

7 Conclusion

7.1 Résumé des principaux résultats

Cette étude s'est déroulée en trois étapes : tout d'abord, j'ai identifié les fonctions spécifiques de chaque postposition en me basant sur le système de classification développé par le projet Sejong et sur les études déjà effectuées sur les postpositions adverbiales Coréennes. La postposition *-ey* a huit fonctions majeures, avec 'localisation' et 'but' qui occupent la majorité des occurrences. *-eyse* a deux fonctions, 'source' et 'localisation', et est utilisé beaucoup plus fréquemment que les autres. *-(u)lo* a six fonctions, dont les trois principales, 'état final', 'instrument' et 'direction', occupent plus de 80% de l'utilisation totale.

Ensuite, j'ai créé les modèles de classification/visualisation, l'un en utilisant une combinaison de PPMI et SVD comme modèle basé sur le nombre et l'autre en utilisant SGNS comme modèle basé sur la prédiction avec la base de l'estimation basée sur la similarité. En général, j'ai constaté que, si une postposition avait moins de fonctions, le modèle de classification obtenait une précision de classification élevée. Le modèle PPMI-SVD a atteint une grande précision de classification lorsque la taille du Window était grande, ce qui indique que pour la meilleure performance de classification, il a utilisé les caractéristiques sémantiques des grandes tailles de Window plus que les caractéristiques syntaxiques. En revanche, le modèle SGNS a montré une faible précision de classification, quelle que soit la taille des Windows. Par ailleurs, j'ai constaté que le modèle PPMI-SVD était plus affecté par la taille du corpus que le modèle SGNS. Cela s'explique par le fait que le modèle PPMI-SVD est sensible à la fréquence des tokens des mots, alors que le modèle SGNS est sensible à la fréquence des types de mots. À travers la visualisation, j'ai trouvé que (i) les clusters n'ont pas changé considérablement par les environnements des vecteurs du niveau du mot, et (ii) il y avait les deux types de mots co-occurrents : les mots qui apparaissaient fréquemment dans le corpus total et les mots qui apparaissaient seulement quand la postposition était utilisée comme une fonction spécifique.

Finalement, j'ai appliqué BERT pour *transformer* tous les mots en différents vecteurs, tout en considérant leurs informations contextuelles pour la même tâche de classification. Pour la tâche de classification, le modèle BERT a obtenu une grande précision de classification : 0.815 pour *-ey*, 0.898 pour *-eyse*, 0.813 pour *-(u)lo*. Ce résultat était supérieur aux performances des modèles des études précédentes et des modèles de vecteur du niveau du mot que j'ai utilisés. En outre, j'ai constaté que le modèle BERT n'était pas particulièrement influencé par la taille du corpus de chaque fonction, contrairement au résultat montré par les modèles de vecteur au niveau du mot. Les raisons en sont que le modèle BERT a attribué à chaque mot un vecteur basé sur les informations contextuelles et a fonctionné sur la base du modèle pré-entraîné avec une grande quantité de données de corpus. A travers la visualisation, j'ai constaté que le modèle BERT pouvait reconnaître les fonctions de chaque postposition au fur et à mesure que l'epoch (i.e., l'apprentissage) progressait, même si les fonctions occupaient une plus petite partie de la taille totale du corpus. Ceci était également contradictoire avec les résultats des modèles traditionnels de vecteur au niveau du mot, qui sont connus pour être considérablement affectés par la taille du corpus. Cela indique que le modèle BERT peut identifier des fonctions relativement peu fréquentes à un niveau satisfaisant grâce à un nombre suffisant d'epochs. D'ailleurs, cela suggère qu'il est capable de simuler la façon dont

les humains interprètent la polysémie impliquant les postpositions adverbiales Coréennes de façon plus appropriée que les modèles de vecteur au niveau du mot.

7.2 Limites et travaux futurs

Malgré ces résultats, cette thèse reste limitée. Je reconnais certaines limites de ce projet comme suit.

Avant tout, je me suis concentré uniquement sur trois différentes postpositions adverbiales Coréennes qui ont une polysémie du niveau des mots. Cependant, selon la description statistique du dictionnaire *Standard-Korean* (1999), il existe 361 postpositions en langue Coréenne. Dans cette optique, les résultats obtenus à partir des trois postpositions étudiées dans cette thèse ne sont pas suffisants pour généraliser toutes les postpositions de la langue Coréenne. Par conséquent, à l'avenir, j'améliorerais cette étude pour couvrir davantage de postpositions qui ont des degrés de polysémie similaires à ceux de *-ey*, *-eyse*, et *-(u)lo*.

Deuxièmement, j'ai utilisé trois modèles de vecteur (PPMI-SVD, SGNS et BERT) pour la tâche de classification dans cette thèse. Cependant, compte tenu du fait que d'autres modèles de vecteur du mot contextualisés ont été publiés après BERT, tels que le *Generation Pre-trained Transformer 3* (GPT-3 ; Brown et al., 2020) ou le *Robustly Optimized BERT Pretraining Approach* (RoBERTa ; Liu et al., 2019b), il est nécessaire de les utiliser afin d'assurer la généralisabilité méthodologique et d'attester des méthodes de calcul récentes en Coréen, une langue typologiquement différente des principales langues indo-européennes.

7.3 Implications des résultats

Malgré ces limites, cette thèse a deux implications majeures.

Premièrement, il fournit les moyens possibles et les limites de l'application de trois modèles de vecteur différents pour la tâche d'identification de la fonction prévue des postpositions adverbiales Coréennes. De nombreuses études ont été réalisées sur l'interprétation de la polysémie au niveau du mot dans les principales langues indo-européennes en utilisant des modèles de vecteur au niveau du mot ou des modèles de vecteur du niveau des phrases dans le cadre de la modélisation sémantique distributive. Bien que de nombreuses recherches aient été menées sur l'anglais à ce sujet, très peu d'études ont porté sur l'interprétation de la polysémie dans une langue typologiquement différente de l'anglais. J'ai donc porté mon attention sur le Coréen, une langue peu explorée à cet égard, en me concentrant sur la relation entre les trois différents modèles de vecteur et la polysémie au niveau du mot des postpositions adverbiales. Considérant que les recherches antérieures sont orientées vers les principales langues indo-européennes telles que l'anglais, la tentative de cette thèse contribue à la généralisation méthodologique en appliquant le calcul à une langue moins étudiée comme le Coréen.

En deuxième lieu, cette thèse propose deux systèmes de visualisation interactifs qui aident à identifier les relations entre les mots ou les phrases et à montrer les changements des groupes en fonction des environnements (i.e., les modèles, les postpositions, la taille des Windows et les epochs). Bien que les modèles de vecteur au niveau du mot et des phrases aient été fréquemment utilisés dans les études récentes, il est très difficile de comprendre comment ces modèles de vecteur interprètent la polysémie au niveau du mot. Le premier système de visualisation visait à explorer les résultats du vecteur au niveau du mot. Cela nous permet de voir les groupes de postpositions et leurs mots co-occurrents afin de comprendre comment les relations entre les mots ont changé en fonction des fonctions de chaque postposition. Le deuxième système de visualisation a été développé pour montrer comment le modèle de vecteur au niveau de la phrase (i.e., BERT) reconnaît la polysémie impliquant les postpositions. Considérant que le système de visualisation pourrait aider à comprendre les résultats de calcul plus facilement et plus clairement grâce à un affichage intuitif (et aussi informatif) des données linguistiques, l'essai de cette

thèse a une contribution particulière pour les études futures.

Bibliographie

Standard Korean Dictionary. National Institute of Korean Language, Seoul, South Korea, 1999.

Myung-chel Ahn. The meaning of locative postposition '*-ey*'. *kwanak emwun yenkwu*, 7 :245–268, 1983.

Jangseong Bae and Changki Lee. End-to-end learning of korean semantic role labeling using bidirectional lstm crf. In *Proc. of the KIISE Korea Computer Congress*, pages 566–568, 2015.

Jangseong Bae, Junho Oh, Hyunsun Hwang, and Changki Lee. Extending korean propbank for korean semantic role labeling and applying domain adaptation technique. *Korean Information Processing Society*, pages 44–47, 2014.

Jangseong Bae, Changki Lee, and Soojong Lim. Korean semantic role labeling using deep learning. *Korean Information Science Society*, 6 :690–692, 2015.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 1 : 238–247, 2014. URL https://www.researchgate.net/publication/270877599_Don%27t_count_predict_A_systematic_comparison_of_context-counting_vs_context-predicting_semantic_vectors.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

John A. Bullinaria and J. Levy. Extracting semantic representations from word co-occurrence statistics : A computational study. *Behavior Research Methods*, 39 :510–526, 2007.

Jeong-mi Cho and Gil-cheng Kim. A study on the resolving of the ambiguity while interpretation of meaning in korean. *The Korean Institute of Information Scientists and Engineers*, 14(7) :71–83, 1996.

Miho Choo and Hye-young Kwak. *Using Korean*. Cambridge University Press, New York, NY, 2008.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1) :22–29, 1989.

Andy Clark. Embodied prediction. In *In T. K. Metzinger J. M. Windt (Eds.) Open MIND : 7(T). Frankfurt am Main : MIND Group*, 2015.

Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. *Comput. Speech Lang.*, 9(2) :123–152, 1995. URL http://dblp.uni-trier.de/db/journals/csl/csl9.html#DaganMM95.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1 : 211–218, 1936.

Katrin Erk. Vector space models of word meaning and phrase meaning : A survey. *Lang. Linguistics Compass*, 6(10) :635–653, 2012. URL http://dblp.uni-trier.de/db/journals/llc/llc6.html#Erk12.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi : 10.18653/v1/D19-1006. URL https://www.aclweb.org/anthology/D19-1006.

J. Firth. *A synopsis of linguistic theory 1930-1955*. Studies in Linguistic Analysis, Philological. Longman, 1957.

Dylan Glynn and Justyna Robinson. *Corpus Methods for Semantics*. Corpus Methods for Semantics. Quantitative studies in polysemy and synonymy. John Benjamins, January 2014. doi : 10.1075/hcp.43. URL https://halshs.archives-ouvertes.fr/halshs-01284061.

Zellig Harris. Distributional structure. *Word*, 10(2-3) :146–162, 1954. doi : 10.1007/978-94-009-8467-7_1. URL https://link.springer.com/chapter/10.1007/978-94-009-8467-7_1.

Yunpyo Hong. On the case of directionality. *The Society of Korean Linguistics*, 6 :111–132, 1978.

Heewon Jeon, Donggeon Lee, and Jangwon Park. Korean bert pre-trained cased (kobert), 2019. URL https://github.com/SKTBrain/KoBERT.

Byong-cheol Jeong. An integrated study on the particle '*-ey*' based on the simulation model. *The Linguistic Science Society*, 55 :275–304, 2010.

Daniel Jurafsky and James. H. Martin. *Speech and language processing : An Introduction to Natural Language Processing*. Computational Linguistics, and Speech Recognition, Prentice-Hall, 2019.

Sin-jae Kang and Jung-hye Park. Rule construction for determination of thematic roles by using large corpora and computational dictionaries. *Korean Information Processing Society*, 10(2) :219–228, 2003.

Byoung-soo Kim, Yong-hun Lee, Seung-hoon Na, Jun-gi Kim, and Jong-hyeok Lee. Bootstrapping for semantic role assignment of korean case marker. *Korea Information Science Society*, pages 4–6, 2006.

Byoung-soo Kim, Yong-hun Lee, and Jong-hyeok Lee. Unsupervised semantic role labeling for korean adverbial case. *Journal of KIISE : Software and Applications*, 34(2) :32–39, 2007.

Wan-su Kim and Cheol-young Ock. Korean semantic role labeling using case frame and frequency. *The Korean Institute of Information Scientists and Engineers*, 6 :651–653, 2015.

Wan-su Kim and Cheol-young Ock. Korean semantic role labeling using case frame dictionary and subcategorization. *The Korean Institute of Information Scientists and Engineers*, 43(12) :1376–1384, 2016.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

Changki Lee, Soojong Lim, and Hyunki Kim. Korean semantic role labeling using structured svm. *The Korean Institute of Information Scientists and Engineers*, 42(2) :220–226, 2015.

Namsun Lee. In the form '*-ey*' and the material '*-eyse*'. *kwanak emwun yenkwu*, 8 :321–355, 1983.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. 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)*, pages 1073–1094, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi : 10.18653/v1/N19-1112. URL https://www.aclweb.org/anthology/N19-1112.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized bert pretraining approach, 2019b.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov) :2579–2605, 2008. ISSN ISSN 1533-7928. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Gyeongheum Maeng. Cognitive semantics of korean postposition '*-ey*'. *The Journal of Korean Studies*, 41 :325–366, 2016.

Chris McCormick. Bert fine-tuning tutorial with pytorch, 2019. URL http://mccormickml.com/2019/07/22/BERT-fine-tuning/.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

Seongmin Mun and Gyu-Ho Shin. Context window and polysemy interpretation : A case of korean adverbial postposition *-(u)lo*. In *IMPRS Conference 2020 : Interdisciplinary Approaches to the Language Sciences, Max Planck Institute for Psycholinguistics*, 2020.

Ki-sim Nam. The use of the korean postposition : focus on '*-ey*' and '*-(u)lo*'. *sekwang hakswul calyosa*, 1993.

Jeong-woon Park. A polysemy network of the korean instrumental case. *Korean Journal of Linguistics*, 24(3) :405–425, 1999.

Seong-bae Park, Byoung-tak Zhang, and Yungtaek Kim. Decision tree based disambiguation of semantic roles for korean adverbial postpositions in korean-english machine translation. *The Korean Institute of Information Scientists and Engineers*, 27(6) :668–677, 2000.

Tae-ho Park and Jeong-won Cha. Korean semantic role labeling using word sense. *The Korean Institute of Information Scientists and Engineers*, 6 :590–592, 2017.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018. URL http://arxiv.org/abs/1802.05365. cite arxiv :1802.05365Comment : NAACL 2018. Originally posted to openreview 27 Oct 2017. v2 updated for NAACL camera ready.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

G. Salton. *The SMART Retrieval System : Experiments in Automatic Document Processing*. Prentice Hall, 1971.

Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases : The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2) : 169–194, jun 1998. URL http://dx.doi.org/10.1023/A:1009745219419.

Hyo-pil Shin. The 21st sejong project : with a focus on selk (sejong electronic lexicon of korean) and the knc (korean national corpus). In *In The 3rd International Joint Conference on Natural Langauge Processing*, 2008.

Myung-chul Shin, Yong-hun Lee, Mi-young Kim, You-jin Chung, and Jong-hyeok Lee. Semantic role assignment for korean adverbial case using sejong electronic dictionary. *Korea Information Science Society*, pages 120–126, 2005.

Ho-Min Sohn. *The korean language*. Cambridge University Press, Cambridge, UK, 1999.

Dae-heon Song. A study on the adverbial case particles of '*-ey*' and '*-eyse*' for korean language education. *The Association of Korean Education*, 101 :457–484, 2014.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136, 2019. URL http://arxiv.org/abs/1903.12136.

Peter D. Turney and Patrick Pantel. From frequency to meaning : Vector space models of semantics. *CoRR*, abs/1003.1141, 2010. URL http://arxiv.org/abs/1003.1141.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers : State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.