



#### Université Paris Nanterre

École doctorale 139 – Connaissance, Langage, Modélisation

# Polysemy resolution with word embedding models and data visualization: the case of adverbial postpositions *-ey*, *-eyse*, and *-(u)lo* in Korean

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

# Acknowledgements

There are many who helped me along the way on this journey.

### **Abstract**

This dissertation reports computational accounts of resolving word-level polysemy in a lesser-studied language-Korean. Postpositions, which are characterized as multiple form-function mapping and thus polysemous in nature, pose a challenge to automatic analysis and model performance in identifying their functions. In this project, I enhance the existing word-level embedding classification models (Positive Pointwise Mutual Information and Singular Value Decomposition; Skip-Gram and Negative Sampling) with the consideration of context window, and introduce a sentence-level embedding classification model (Bidirectional Encoder Representations from Transformers (BERT)) under the scheme of Distributional Semantic Modeling. I then develop two visualization systems that show (i) relationships of the postpositions and their co-occurring words for word-level embedding models, and (ii) clusters between sentences for the sentence-level embedding model. These visualization systems have an advantage to better understand how these classification models classify the intended functions of these postpositions. Results show that, whereas the performance of the word-level embedding models is modulated by the size of training corpora containing specific functions of the postpositions, the sentence-level embedding model performs

in a stable way (i.e., less affected by the corpus size) and simulates how humans recognize the polysemy involving Korean adverbial postpositions more appropriately than the word-level embedding models do.

**Keywords:** polysemy, natural language processing, classification, word embedding models, data visualization, Korean

# **Contents**

1	Intro	oduction	1
	1.1	Background of beginning this project	1
Α	Cod	e for the word-level embedding models	3
Re	fere	nces	5

# **List of Tables**

# **List of Figures**

## List of abbreviations

The following abbreviations are used to label the linguistic terms employed in this dissertation. I follow the Leipzig glossing rules<sup>1</sup> for the most abbreviations used in linguistic glosses.

<sup>&</sup>lt;sup>1</sup>Available at: https://www.eva.mpg.de/lingua/pdf/Glossing-Rules.pdf

Label
Accusative
Agent
Content
Comitative
Criterion
Declarative
Direction
Effector
Experiencer
Final State
Goal
Indicative
Instrument
Location
Mental Agent
Nominative
Plural
Present
Past
Purpose
Source
Theme

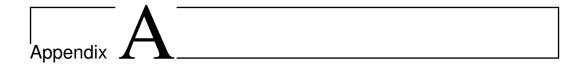
1	
Chapter	

## Introduction

The project presented in this dissertation aims to address the possible ways and limitations in applying computational approaches to word-level polysemy in a lesser-studied language, Korean.

## 1.1 Background of beginning this project

I assume that a relationship of words (represented as probabilistic information) is one core construct in understanding how language works.



# Code for the word-level embedding models

The following scripts are the code that I used for the training of *traditional* word embedding models (i.e., PPMI-SVD, SGNS) and *similarity-based* estimation.

Listing A.1: Python code for the word embedding by using the PPMI-SVD model

```
1
2 class PPMI_SVD_Algorithm:
3
4
      def __init__ (self, fold, postposition, postposition_ko,
          window):
5
          self.fold = fold
          self.postposition = postposition
6
7
          self.postposition_ko = postposition_ko
8
          self.window = window
9
10
      def PPMI_SVD_Calculation(self):
11
12
          from collections import Counter
13
          import itertools
```

14	<pre>import nltk</pre>
15	<pre>from nltk.corpus import stopwords</pre>
16	<pre>import numpy as np</pre>
17	import pandas as pd
18	<pre>from scipy import sparse</pre>
19	<pre>from scipy.sparse import linalg</pre>
20	<pre>from sklearn.preprocessing import normalize</pre>
21	<pre>from sklearn.metrics.pairwise import cosine_similarity</pre>

# References