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

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[1] for the most abbreviations used in linguistic glosses.

[1]Available at: https://www.eva.mpg.de/lingua/pdf/Glossing-Rules.pdf

Abbreviation	Label
ACC	Accusative
AGT	Agent
CNT	Content
COM	Comitative
CRT	Criterion
DECL	Declarative
DIR	Direction
EFF	Effector
EXP	Experiencer
FNS	Final State
GOL	Goal
IND	Indicative
INS	Instrument
LOC	Location
MAG	Mental Agent
NOM	Nominative
PL	Plural
PRS	Present
PST	Past
PUR	Purpose
SRC	Source
THM	Theme
TOP	Topic

Chapter 1

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.

Appendix A

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

```

class PPMI_SVD_Algorithm:

    def __init__ (self, fold, postposition, postposition_ko,
        window):
        self.fold = fold
        self.postposition = postposition
        self.postposition_ko = postposition_ko
        self.window = window

    def PPMI_SVD_Calculation(self):

        from collections import Counter
        import itertools
```

```python
import nltk
from nltk.corpus import stopwords
import numpy as np
import pandas as pd
from scipy import sparse
from scipy.sparse import linalg
from sklearn.preprocessing import normalize
from sklearn.metrics.pairwise import cosine_similarity
```

References