Using Translations to Distinguish Between Homonymy and Polysemy

by

Amir Ahmad Habibi

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Distinguishing between homonymy and polysemy can facilitate word sense disambiguation (WSD), as WSD systems use the standard sense inventories that are excessively fine-grained and include many polysemous senses. We classify words as either homonymous or polysemous by building graphs of word senses as nodes and the semantic relatedness between them as edges. To find these edges, we build upon a previous hypothesis on the role of shared translations in semantic relatedness, and after developing new theorems, we propose methods of obtaining semantic relatedness information from translations of multilingual sense inventories. Additionally, we create a new homonym resource, as well as updating an existing one. These resources include lists of homonyms and their short definitions, manually mapped to their corresponding WordNet senses. We evaluate our methods on a balanced dataset of homonymous and polysemous words, and our methods achieve state-of-the-art results in the task of homonym detection.

Preface

The research presented in this dissertation is a joint work with Prof. Grzegorz Kondrak. The author contributed ideas, implemented the methods, and performed the experiments. The type-A homonym resource presented in this work is the result of a collaborative effort of the NLP group at the University of Alberta, including Bradley Hauer and Yixing Luan. The author was the main contributor, performing the majority of the manual annotations. The type-B homonym resource presented in this work is constructed by the author.

The author has also co-authored three shared-task papers during the program at the University of Alberta (Hauer et al., 2019, 2020a,b).

Acknowledgements

I wish to express my gratitude to my supervisor, Professor Greg Kondrak, for his guidance and support.

I must also thank Bradley Hauer for his help and suggestions. And I am also grateful to our NLP group. Thank you Yixing, Rashed, Arnob, and Hongchang.

Finally, I have to thank my parents who have always been there for me.

This research was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, and Alberta Advanced Education.

Contents

List of Tables

List of Figures

Glossary

cognate
: A word that comes from the same origin as another word in another language.

concept
: A discreet meaning that can be represented by a synset in a sense inventory.

holonymy
: The relation of Y to X, where X is a part/member of Y.

homograph
: A word that has different pronunciations depending on its sense.

homonymy
: The relation between senses that are semantically unrelated, as opposed to polysemy.

hypernymy
: The relation of Y to X, where X is a Y.

hyponymy
: The relation of X to Y, where X is a Y.

lemma
: A lemma or citation form is a grammatical form of a word representing it in a dictionary.

lexeme
: A pair of word form and its meaning.

meronymy
: The relation of X to Y, where X is a part/member of Y.

parallel homonymy
: The phenomenon of two homonymous words lexicalizing the same two semantically unrelated concepts.

parallel corpus
: A collection of texts, translated into one or more languages.

polysemy
: The relation between senses that are semantically related, as opposed to homonymy.

sense
: A discrete representation of a meaning of a word.

sense inventory
: A computational resource containing senses of words in synsets.

synonymy
: The relation of sameness of meanings.

synset
: A group of synonymous senses in a sense inventory representing a concept.

type-B homonym
: A homonymous word that emerged when a word obtained a new meaning over time.

type-A homonym
: A homonymous word that originated as distinct words converging into the same orthographic form.

word sense disambiguation (WSD)
: The task of choosing a sense for a word token, from a sense inventory.

Chapter 1

Introduction

This chapter, first, defines the problem of homonym detection and discusses its importance and challenges. Then our method of homonym detection is described along with our additional contributions in creating homonym resources.

1.1 Problem

Homonymy is the relation between semantically unrelated senses of a word (Jurafsky and Martin, 2008). For example, the word *bank* has the senses of *sloping land* and *financial institution* with the homonymy relation. Words with semantically unrelated senses, such as *bank*, are called homonymous. On the other hand, the relation between semantically related senses of a word is that of polysemy, and words with multiple but semantically related meanings are called polysemous. For example, *debt* is a polysemous word, with the meanings of *the state of owing* and *the things owed*. Furthermore, we use the term *lexeme* to refer to a pair of word and meaning in a lexicon (Jurafsky and Martin, 2008). Each lexeme covers a set of semantically related senses of a word. Therefore, a homonymous word has more than one lexeme, where a polysemous word has only one lexeme covering all of its senses. Table 1.1 demonstrates the above examples, along with their lexemes and senses. In this table, the parts of speech (POS) of the words are shown in subscripts and the sense numbers are in superscripts.

The primary objective of this thesis is to distinguish between homonymous

Word	Lexemes	Senses
$bank_n$	1. ridge	$bank_n^1$: Sloping land
		$bank_n^3$: Long ridge or pile
		$bank_n^4$: Objects in a row
	2. repository	$bank_n^2$: Financial institution
		$bank_n^5$: Supply held for future
$debt_n$	1. owing	$debt_n^1$: State of owing
		$debt_n^2$: Things owed
		$debt_n^3$: Obligation to pay

Table 1.1: Examples of words and their senses grouped under their lexemes, where the homonymous word, $bank_n$, has more than one lexeme.

and polysemous words, given their senses. More specifically, when classifying a word as either homonymous or polysemous, the input is a list of its senses and their translations. Furthermore, achieving the above classification of a word requires distinguishing between the homonymy and polysemy of the *senses* of that word. Therefore, the output is, first, the binary classification of each pair of senses of the word as homonymous or polysemous and, second, the classification of the word itself. We should note that the words with only one sense (monosemes) are not considered in this task, as they do not belong to either of the two classes, and their classification is trivial.

The task of distinguishing between homonymy and polysemy has its own challenges. The most challenging cases of homonymy detection are those almost similar senses that even human agents cannot always agree on their relatedness. For example, the English word $bang_n$ has two lexemes, one with the meaning of *sudden noise* and another related to *narcotics*. WordNet, on the other hand, includes the sense $bang_n^3$, which is defined as "*A border of hair that is cut short and hangs across the forehead*". Based on the definitions, there is no apparent relation between the above sense and any of the two lexemes. Nevertheless, if we investigate the history of this sense, we would realize that it comes from the term *bang-off*, which originally meant *to cut suddenly*. Then we can make the connection to the first lexeme that means *sudden noise*. On the other hand, one can argue that this new meaning has developed a new lexeme, and it is no longer semantically related to any of the two lexemes men-

tioned above. This argument demonstrates one of the challenges embedded in our task.

1.2 Motivation

The study of homonymy can facilitate natural language processing (NLP) tasks such as word sense disambiguation (WSD) and machine translation, as homonymy is important in building the set of senses of a word (Hauer and Kondrak, 2020a). In building sense inventories, which are large collections of words and their sets of senses, the level of granularity can vary. This level of granularity can be an important factor in tasks such as WSD that use a sense inventory. In WSD, which is the task of choosing a sense for an instance of an ambiguous word, using a more fine-grained sense inventory adds to the challenge. Considering the example of $bank_n$ in Table 1.1, we can observe that, even for a human judge, it is more challenging to choose between the fine-grained senses of $bank_n^1$ and $bank_n^3$ belonging to the same lexeme, rather than choosing between $bank_n^1$ and $bank_n^2$ that belong to different lexemes. Therefore, a coarser granularity at the level of lexemes can simplify the process of WSD.

Despite the fact that fine granularity of senses adds to the challenges of some NLP tasks, this issue is still present in Princeton WordNet[1], which has been a standard sense inventory in NLP for a long time (Navigli, 2018). As an example of this problem, we can mention the word $camp_n$ with eight WordNet senses. Among these eight senses are *army camp*, *travelers camp*, and *children summer camp*, demonstrating a very fine-grained distinction. This fine granularity in WordNet has been a challenge for NLP tasks, such as WSD, since the 1990s and early 2000s, during which WSD systems could barely reach 65% accuracy (Navigli, 2018). This issue has also been a challenge for *human* sense annotators. A report demonstrates an inter-annotator agreement of 72% for this task using the fine-grained WordNet senses (Navigli, 2006), which can signify the importance of having a wordnet that differentiates between senses in a coarser level. Homonymy and polysemy can be seen as an objective criterion

[1]Often referred to as just WordNet.

to cluster similar senses in sense inventories and fix the problem of fine granularity. However, solving the problem of WordNet, would not be a one-time task as there are sense inventories in different languages that are usually of a lower quality than the English Princeton WordNet, which can benefit from an automatic method to reduce the granularity of senses. Moreover, as the languages change and new concepts are added to sense inventories, the importance of having an automatic method of fixing the problem of fine granularity would be more evident. For example, Wiktionary is a sense inventory that is updated continuously and can benefit from an automatic method of sense clustering.

1.3 Prior Work

In the prior work, we have found two main approaches when encountering the fine-granularity of WordNet. One approach is to cluster WordNet's similar senses, which has been attempted both manually and automatically (Navigli, 2006; Hovy et al., 2006). The other approach is to automatically define new sense inventories through translations (Dyvik, 2004, 2009; Van der Plas and Tiedemann, 2006). These projects have inspired us in the development of our methods of semantic relatedness and homonymy detection. Furthermore, there has been other work on homonyms, discussing their distributional properties and presenting hypotheses about them (Beekhuizen et al., 2018; Rice et al., 2019; Hauer and Kondrak, 2020a). Moreover, van den Beukel and Aroyo (2018) has worked specifically on the problem of distinguishing between homonymy and polysemy. A further discussion of the related work is in Chapter 3.

1.4 Thesis Statement

The main thesis of this work is: *words can be classified as either homonymous or polysemous using graphs constructed from translation information in multilingual wordnets.*

1.5 Methods

Our method uses translations as evidence for semantic relatedness. We build upon the One Homonym Per Translation (OHPT) hypothesis of Hauer and Kondrak (2020a), which states that a shared translation is a sufficient condition for polysemy. Afterward, by adding further theorems of semantic relatedness, we obtain semantic relation information from translations in multilingual wordnets, and try to distinguish between homonymy and polysemy.

The first step of our method, for each word, is to consider all pairs of its senses and try to find evidence for their semantic relatedness. For example, the two senses of *debt*, with the meanings of *the state of owing* and *the things owed*, are both translated by the Greek word χρέος. One of our semantic relatedness methods would consider this translation as evidence for the polysemy of the two senses of *debt*.

In the second step, we build a graph of senses for each word with edges that indicate semantic relatedness between the senses. We assume this relation to be transitive, as it is intuitive to consider any two senses that are both semantically related to a third sense, to be semantically related to each other as well. For example, in Table 1.1, if we knew that $bank_n^1$ is related to $bank_n^3$ and $bank_n^3$ is related to $bank_n^4$, we would assume that $bank_n^1$ is related to $bank_n^4$. This transitivity in our graph means two senses are semantically related if there is a path of any length between them. Therefore, our final decision on the classification of a word into homonymous or polysemous depends on the number of connected components of this graph. In other words, if a word has distinct groups of senses with no semantic relatedness between them, we consider that word homonymous. Otherwise, the word is classified as polysemous.

1.6 Contributions

Employing formerly and newly introduced theories, this thesis demonstrates their performance in the application on a larger scale and achieves state-of-

the-art results in the task of homonym detection. Our best method achieved an F_1 score of 79% with a precision of 74% and a recall of 83%.

Apart from the methods of homonym detection, this work presents two homonym resources, one created and another updated, both manually. Both of these resources include homonymous words and a mapping from the homonyms to their corresponding senses in WordNet. Our experiments make use of these resources as a part of the gold data and for analysis. However, their application is not limited to our experiments, and they can facilitate future research on homonymy, semantic relations, and sense inventories.

The first resource we provide is an update on the list of type-A homonyms by Hauer and Kondrak (2020a). The definition of type-A and type-B homonyms is in Chapter 2. This list includes 804 lemmas, each with two or more lexemes that are mapped to their corresponding senses in WordNet. Our main contribution to this resource is manually updating and fixing the mapping of more than 600 senses that were either missing or incorrect.

The second resource is an additional list of 94 homonymous words with their lexemes mapped to their corresponding WordNet senses. We originally obtained a list of homonymous lemmas from the work of Rice et al. (2019), and extracted their lexemes from the Wordsmyth dictionary[2]. Afterward, we manually mapped these lexemes to their appropriate WordNet senses. We believe this resource to be mostly consisting of type-B homonyms, as our first list of type-A homonyms is considered a representative of all English type-A homonyms.

1.7 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, we review some of the preliminary concepts and their definitions, along with the resources used in this research. Then, Chapter 3 provides a more detailed discussion of the prior work mentioned in the current chapter. Afterward, Chapter 4 discusses our methods and theorems of finding semantic relatedness between senses and

[2]https://www.wordsmyth.net/

building graphs of semantic relatedness for words. Chapter 5 provides details on our data collection process, our experiments using the collected data and the methods mentioned above, and an error analysis on the results. Chapter 6 concludes this thesis, summarizing the results and discussing possible future directions. In the end, the Glossary contains short definitions of the terms used in the thesis.

Chapter 2

Background

In this chapter, We first discuss the preliminary concepts and terms used throughout this thesis. Then, we move to the resources that we employed in our experiments, describing their features and presenting statistics about them.

2.1 What is Homonymy?

A word might have unrelated meanings, as in *bank*, with the meanings of *repository* and *ridge*. These meanings of a word are called homonyms. However, this is a crude definition, and in order to have a more formal definition, we would first define some preliminary terms, including lexeme, lemma, and sense.

Lexemes are pairs of words and meanings, and a list of lexemes would create a lexicon (Hauer and Kondrak, 2020a). For example, *bank* and its *repository* meaning constitute a lexeme that would have an individual entry in a dictionary, where *bank* and its *ridge* meaning constitutes another lexeme with a separate entry.

Moreover, *words* are sets of wordforms representing lexemes (Hauer and Kondrak, 2020a). We consider words with different parts of speech (POS) to be distinct. Therefore, the noun $bank_n$ and the verb $bank_v$ are different words to be analyzed individually.

A **lemma** is the canonical form of a wordform representing a lexeme in a lexicon or dictionary (Jurafsky and Martin, 2008). For example, wordforms

such as *tending* and *tended* are both represented by the lemma *tend*. Different wordforms with the same lemma are considered the same; therefore, throughout our experiments, we work with only lemmas and not the other wordforms.

Senses are discrete representations of different meanings of a word (Jurafsky and Martin, 2008). We represent the senses, throughout this thesis, using a superscript over the lemma. Therefore, in the above example, we can have $tend^1$ and $tend^2$ as senses of the lemma *tend*.

Concepts are meanings that when lexicalized by lemmas establish senses. Each concept can be represented by one or more lemmas and each lemma can be paired with one or more concepts (Hauer and Kondrak, 2020b). As *synonymy* is the relation of sameness of meaning, the lemmas representing the same concept are synonyms, and they would create synonym sets or *synsets*. Consequently, each synset belongs to a concept. For example, the concept of *a motor vehicle with four wheels* is lexicalized by lemmas such as *car*, *automobile*, and *machine*. These three lemmas, paired with the above concept, create three senses that would form a synset.

Considering the above definition of senses, we can define **homonymy**, which is the relation between those senses of a word that are semantically unrelated, where their similar orthographic form might even be a coincidence (Jurafsky and Martin, 2008). Our example of *tend* has two seemingly unrelated senses of *having a tendency to* ($tend^1$) and *caring for* ($tend^2$). These senses are referred to as homonyms of each other, and the relation between them is that of homonymy. On the other hand, semantically similar senses of a word have the relation of **polysemy**. For example, *tend* has another sense with the meaning of *managing or running* ($tend^3$), which is semantically related to *caring for* ($tend^2$). The polysemous senses of a word belong to the same lexeme. Therefore, if a word has more than one lexeme, it would be considered homonymous (Hauer and Kondrak, 2020a). Having these general definitions, we should answer three more detailed questions about our definition of homonymy.

The first question is whether or not to consider different parts of speech

(POS) of the same word as different words. For example, the word *bear* has two distinct meanings of *carry*, a verb, and *animal*, a noun. If we consider the noun *bear* ($bear_n$) and the verb *bear* ($bear_v$) to be two different words, they would be two non-homonymous words. However, if we view $bear_n$ and $bear_v$ as different types of the same word, then that single word would be homonymous. There exist arguments for both of these views. For the first view, one could argue that the POS of a word in a sentence is almost always clear, using the syntax of the sentence. Therefore, in practice, there would be no ambiguity introduced in the sentence by the use of that word. On the other hand, we can argue that it is possible to transfer the meanings of a word across parts of speech. In our example of $bear_n$, the meaning of *animal* can be used as a verb with a related meaning, such as *acting like a bear*. As a result, the POS would become irrelevant, and we can claim that different POS of a word should not constitute different words. Nonetheless, there is still a debate going on in Cognitive Science studies on whether or not nouns and verbs are represented separately in our mental semantic space (Mirman et al., 2010). Our experiments are based on the first assumption, which is to consider different parts of speech to be different words. We make this assumption because our efforts are towards improving the disambiguation process and resolving the POS can diminish part of the ambiguity.

The next issue is whether or not to view named entities (proper nouns) as separate senses for words. Let us take the example of the word *smith* with the sense of *metal worker*, which is also the name of the economist *Adam Smith*. Considering that the latter is a *named entity* and not a dictionary word, we prefer to view *smith* as non-homonymous. A reason for this approach is that named entities are not in the dictionaries such as Oxford. Therefore, based on those resources, named entities do not constitute a new meaning for the word. Since our gold standard resources are built based on dictionaries, we cannot consider named entities as additional meanings for words.

Finally, there is the issue of inflected forms that are shared with other words and whether or not to consider them a homonym. Consider the example of *ground* with the meanings of *land* and the past tense of *grind*. We could

either consider *ground* as a homonymous word with two meanings or as a non-homonymous word with only the meaning of *land*. In order to resolve this issue, we turn to our sense inventories, which are computational resources containing different senses of words along with their definitions. We are limited by our sense inventories in the way that we can only consider the words that are present in them. However, as sense inventories are similar to dictionaries in having only the canonical form of the words, which are the lemmas, we would not consider inflected forms to be constituting new homonyms.

So far, we have delineated our criteria for homonymy. However, regardless of the details of our definitions, homonyms can be classified into type-A and type-B based on how they are originated (Hauer and Kondrak, 2020a). Type-A homonyms are distinct words converged into the same orthographic form. For example, the Latin word *data* for "given or delivered" and the Greek word *daktulos* for "finger" both converged to the modern English word *date*. The latter developed the meaning of "day" and the former the meaning of "fruit". On the other hand, type-B homonyms would emerge when a word obtains a new meaning over time, developing a new lexeme. For example, the two meanings of "employee" and "stick" of the word *staff* both come from the same etymology of *stæf* in old English. It is important to note that this classification would not affect our definition of homonymy or our methods of homonym detection.

An additional term that we need to mention is *homograph*. Homographs are words that have different pronunciations depending on their sense(Gorman et al., 2018). The list of homographs and homonyms have overlaps in many cases. The word *bow* is an example of such cases, where its homonymous senses of *weapon* and *bend* have distinct pronunciations. However, some homographs are not homonyms. For example, the word *read* has different pronunciations for its past and present tenses, yet it is not a homonym as it only covers one lexeme.

2.2 More About Sense Inventories

Sense inventories such as WordNet (Fellbaum, 1998) and BabelNet (Navigli and Ponzetto, 2012) are large collections of *concepts*. In WordNet and BabelNet, each concept has a definition and a set of lemmas carrying the meaning of that concept. These lemmas form a synonym set or **synset**. As mentioned before, a lemma paired with a concept[1], is called a sense. Therefore, if a lemma has more than one meaning or sense, it should appear in more than one synset. Here is an example of the three concepts in WordNet 3.0 that contain the lemma *twig*, where each synset is represented by its ID[2]:

- *wn:13163991n*:
  - definition: *a small branch or division of a branch (especially a terminal division); usually applied to branches of the current or preceding year*
  - synonyms: *branchlet, twig, sprig*
- *wn:0329654v*:
  - definition: *branch out in a twiglike manner*
  - synonyms: *twig*
- *wn:0590366v*:
  - definition: *understand, usually after some initial difficulty*
  - synonyms: *catch_on, get_wise, get_onto, tumble, latch_on, cotton_on, twig, get_it*

Other sense inventories such as BabelNet have similar features to those that are mentioned in the above example. Further information about each of these sense inventories is presented in section 2.8.

WordNet and BabelNet, include semantic relations such as hypernymy and holonymy, as well as synonymy. The synonymy is embedded in the synsets, as the lemmas in a synset are synonymous with each other. However, hypernymy and meronymy are between concepts. **Hypernymy** is the relation of Y to X,

[1]We use the terms concept and synset interchangeably throughout this thesis as there is a one-to-one correspondence between them.

[2]A WordNet synset ID contains an offset number, unique for each synset, followed by a letter in the end, indicating the POS.

where X *is a* Y, and the relation of X to Y is called **hyponymy** (Jurafsky and Martin, 2008). Furthermore, **holonymy** is the relation of Y to X, where X *is a part/member of* Y, and the relation of X to Y is that of **meronymy**. For example, *plant* is a hypernym of *tree*, and *tree* is a hyponym of *plant*. On the other hand, *forest* is a holonym of *tree*, and *tree* is a meronym of *forest*. Having these relations, a sense inventory can form a graph of hypernymy or holonymy.

Moreover, some sense inventories such as BabelNet are multilingual and can be used to obtain translations of words, based on their senses. These multilingual resources contain lemmas of different languages in their synsets, which are also referred to as multi-synsets (Hauer and Kondrak, 2020b).

It is important to note that even though a dictionary can provide us with sense information as the sense inventories do, they do not provide synsets. Moreover, sense inventories such as WordNet and BabelNet are made particularly for computational use. This makes these sense inventories not only more easily available, but also more compatible with each other, in the case that we require combining their information. For example, we know that synset *wn:13163991n* in WordNet 3.0, corresponds to the synset *bn:00012756n* in BabelNet. However, such information is not available for the Oxford English Dictionary as we have no mapping between WordNet synsets and the OED entries.

Using the above structure of sense inventories, we can inspect the homonymy of words based on the set of senses that are present in the sense inventories. In other words, if a word or a sense does not appear in our target sense inventory, we would not be able to inspect that word or sense. For example, WordNet has a sense of "productive work" for the English word $toil_n$, but it is missing the sense of "trap", which is another meaning of $toil_n$. In this situation, we cannot analyze the relation between the two senses of $toil_n$.

2.3 Parallel Corpora

A source of translation, in addition to multilingual sense inventories, is parallel corpora. A parallel corpus is a collection of texts translated into one or more languages. Many available parallel corpora have their corresponding translated sentences aligned together[3]. From these sentence by sentence translations, word translations can be obtained automatically, using the distributional information of the co-occurrence of the words in sentence pairs. This process is called automatic word alignment.

Before further discussing the parallel corpora, we need to explain the token and type distinction in text corpora. A *word token* is an instance of the usage of a *word type* in some context, where the *word type* is the abstract form of the *word tokens*. Therefore the number of *word types* would be the number of distinct words in a corpus or a vocabulary, and the number of *word tokens* would be the total number of words, including their recurrences in a corpus.

A difference between the translations obtained from a multilingual sense inventory and a word-aligned parallel corpus is that in the former, each translation corresponds to a *sense* of a word type, yet in the latter, each translation corresponds to a word token. To inspect the homonymy of a word using our definition, we would require translations of the senses of that word. Therefore, the word tokens of the corpus are to be mapped to senses of a sense inventory, based on their context. This can be achieved by sense annotation of at least one of the languages of the parallel corpus. An example of a sense-annotated parallel corpus is EuroSense (Delli Bovi et al., 2017).

2.4 Word Sense Disambiguation

Word sense disambiguation (WSD) is the task of assigning a sense to a word token, based on its context. This task is done using sense inventories such as WordNet as the source of senses. For example, in the sentence "She tends to the children", the word *tend* needs to be disambiguated, as it has three

[3]An extensive collection of parallel corpora of many languages can be obtained from OPUS (http://opus.nlpl.eu/).

senses in WordNet: *have a tendency to*, *look after*, and *manage or run*. In this example, WSD is to choose one of these three senses for *tend* in the above sentence. The importance of this task can be particularly seen in translation when a translation equivalent for an ambiguous word is to be chosen, and the first step is to disambiguate the word.

2.5 One Homonym Per Translation

The One Homonym Per Translation (OHPT) hypothesis of Hauer and Kondrak (2020a) is fundamental to our methods of semantic relation detection. This hypothesis states that different lexemes of a word do not share any translations. Here is the formal definition of the hypothesis according to their paper:

Hypothesis 1 (OHPT)**.** *Let $\mathcal{L}$ and $\mathcal{W}$ be the sets of lexemes and words of a language, respectively. Then, let w: $\mathcal{L} \mapsto \mathcal{W}$ be a function mapping the lexemes to their corresponding words. Further, let w^{-1}: $\mathcal{W} \mapsto \mathcal{L}$ be a function mapping the words to their corresponding lexemes. Now, let $T(L)$ be a set of translations of a lexeme L, and let $\mathcal{H}$ be the set of all homonymous words. Then,*

$$\forall H \in \mathcal{H} : \forall L, L' \in w^{-1}(H) : (L \neq L') \Rightarrow T(L) \cap T(L') = \emptyset$$

From this hypothesis, Hauer and Kondrak derive this conclusion:

Corollary 1.1. *The existence of a shared translation is a sufficient condition for polysemy.*

The only actual exception found to this hypothesis is a rare phenomenon that is referred to as *parallel homonymy* (Hauer and Kondrak, 2020a), discussed in section 2.6.

2.6 Parallel Homonyms

Parallel homonymy happens when two unrelated concepts are both lexicalized by the same two homonymous words. These two words can belong to the

same language or two different languages. Since concepts are represented by multi-synsets, we can have the following definition:

Definition 1. *Consider the multi-synsets Syn*1 *and Syn*2 *that are not semantically related. X and Y are parallel homonyms if there exists senses* X^1 *and* Y^1 *in Syn*1*, and* X^2 *and* Y^2 *in Syn*2*.*

Parallel homonymy can happen across languages. One of the sources for parallel homonymy across languages is cognates that arise from the process of lexical borrowing (Hauer and Kondrak, 2020a). Cognates are words in different languages that come from the same etymology. As an example, we can mention the Italian word *banda* that is a parallel homonym with the English word *band*. Both of these words have the two meanings of *a thin strip or loop* and *a group of people*, which are homonymous to each other.

We have also observed parallel homonymy within a language. The cases of parallel homonymy within a language are expected to be rarer than the cases across languages. A common reason for parallel homonymy within a language is spelling variations, such as *mold* and *mould*. However, we have encountered other cases as well. For example, the English words *rig* and *set up* both have these two synsets that are semantically unrelated:

- *bn*:00093035*v* : *Arrange the outcome of by means of deceit*
- *bn*:00093037*v* : *Equip with sails or masts*

2.7 Evaluation Metrics

Our task is to classify words into positive cases of homonymous and negative cases of polysemous. A simple evaluation measure is *accuracy*, which is the number of correct classifications over the number of all cases. However, this measure would not provide insight into how many of the positive classifications were correct or how many of the actual positive cases were detected. To gain this information, we would use additional metrics of precision, recall, and F_1 score. These metrics would use the number of true-positive (TP), true-negative (TN) false-positive (FP), and false-negative (FN) cases in the classifications.

Precision determines the ratio of the cases that were correctly classified as positive:

$$Precision = \frac{TP}{TP + FP}$$

On the other hand, recall determines the ratio of the positive cases detected to all positive cases:

$$Recall = \frac{TP}{TP + FN}$$

Finally, F_1 score is the harmonic mean of precision and recall (Jurafsky and Martin, 2008, p. 745), which would demonstrate a balance between the two measures:

$$\frac{1}{F_1} = \frac{1}{Precision} + \frac{1}{Recall}$$

2.8 Data and Resources

2.8.1 WordNet

WordNet (also called Princeton WordNet[4]) is a manually crafted lexical database Miller et al. (1990). It contains groups of English nouns, verbs, adjectives, or adverbs in synsets (synonym sets). These synsets are connected with different relation links such as antonymy, hyponymy, and meronymy. The synsets contain a short definition of the concept they are representing.

Each word in WordNet can have different meanings that can put it in different synsets, so each occurrence of a word in a synset is an individual object that is called *sense*. The notation for a sense consists of the lemma, the part of speech tag, and the sense number separated by dots. For example, the word *club* with the part of speech noun is in six synsets, which means it has six senses. The first sense, $club_n^1$ has the meaning of *baseball team*, where $club_n^3$ has the meaning of *stick*.

Among the relations between the synsets in WordNet that we have used in our work are hyponymy-hypernymy and meronymy-holonymy. These are defined by Miller et al. (1990) in this way:

[4]We generally call this resource WordNet except when we need to show the contrast with other resources with similar names.

Hypernymy-hyponymy is defined as a semantic relation between word meanings (i.e. senses), and it can be simply expressed as x *is a kind of* y, where x is a hyponym of y and y is a hypernym of x. For example bus_n^4 is a hyponym of car_n^1 and car_n^1 is a hypernym of bus_n^4. Since the senses of each synset represent the same concept, this relation also holds between the synsets that contain the two senses.

Meronymy-holonymy is another semantic relation that is expressed as "*has a*" and "*is part of*" between two synsets. For example, the synset that contains $bumper_n^2$ is a meronym of the synset that has car_n^1, and the latter is a holonym of the former.

WordNet is currently at version 3.0, which is what we used in our experiments. Table 2.1 shows some statistics[5] for this version of WordNet.

WordNet is accessible through its online website[6] and its Application Programming Interfaces (API) for several programming languages, including Python, where it is available as a part of the NLTK library.

POS	Words	Synsets	Senses
Noun	117,798	82,115	146,312
Verb	11,529	13,767	25,047
Adjective	21,479	18,156	30,002
Adverb	4,481	3,621	5,580
Total	155,287	117,659	206,941

Table 2.1: WordNet 3.0 database statistics.

2.8.2 MultiWordNet

MultiWordNet (MWN) is an Italian wordnet that is closely aligned with Princeton WordNet Pianta et al. (2002). It was built by extending the English synsets from WordNet version 1.6 with Italian senses, semi-automatically. Some multi-synsets in MWN have Italian definitions, in addition to the English definitions that come from WordNet. MWN is currently at version 1.5.0,

[5]https://wordnet.princeton.edu/documentation/wnstats7wn
[6]http://wordnetweb.princeton.edu/perl/webwn

which is the version we have used in our work and is available through its website[7] where access to its SQL database files can be requested.

Initially, we reacquired a mapping between MWN to Princeton WordNet to be able to compare the results of our analyses. We first managed to map each MWN synset to its corresponding synset in WordNet 1.6. However, we needed a mapping to the latest WordNet version, which was 3.0. Therefore, we required a mapping of the synsets from WordNet 1.6 to 3.0. Daudé et al. (2000) had made this available in their work, and we were able to obtain an updated version of that work[8]. This mapping is not one-to-one and is probabilistic. However, for the majority of the synsets, the mapping is with the probability of 1.0. Eventually, we were able to create a list of all MWN synsets mapped to WordNet 3.0 synsets.

A piece of additional information provided in MWN is lexical gaps (Bentivogli and Pianta, 2000). Lexical gaps are the concepts that are not represented in a language with a single word, which in this case is Italian. MWN marks some of its synsets with the lemma "GAP" indicating that the synset does not have any single Italian words to represent it. For example, the English word $seamless_a$ has a synset with the meaning of "not having or joined by a seam or seams", and this synset is a lexical gap in Italian, having only a "GAP" lemma in its Italian side.

2.8.3 BabelNet

BabelNet is a large multilingual wordnet that has been built by automatically integrating Princeton WordNet and Wikipedia as well as several other lexical resources in other languages Navigli and Ponzetto (2012). Due to the automatic process of mapping WordNet synsets to Wikipedia pages, the resulting synsets can be noisy, and the F_1 score reported in their first version of BabelNet was %77.7 for their best mapping method Navigli and Ponzetto (2012). BabelNet can be accessed through its website[9] and its APIs, among which the

[7] http://multiwordnet.fbk.eu/

[8] http://www.talp.upc.edu/content/wordnet-mappings-automatically-generated-mappings-among-wordnet-versions

[9] https://babelnet.org/

Java API was the one that we have used.

Currently, BabelNet is at version 4.0 containing 284 languages[10], some of which have wide coverage of the synsets. BabelNet has integrated 47 sources into 15,780,364 synsets in its latest version. Further statistics can be found in Table 2.2.

POS	All Senses	English senses
Noun	807,687,873	22,728,996
Verb	633,627	61,155
Adjective	537,552	112,718
Adverb	115,056	19,653
Total	808,974,108	22,922,522

Table 2.2: BabelNet 4.0 statistics.

The multilingual synsets or multi-synsets in BabelNet contain senses that are tagged with their original resource. For example, BabelNet has a multi-synset with the ID *bn*:00008364*n* has, among others, the senses *WN:EN:bank*, *WIKT:EN:bank*, *WIKI:EN:bank*, *WIKI:IT:banca*, and *IWN:IT:banca*. With each sense, three pieces of information are provided: the original resource, the language, and the lemma. Therefore, in this example, we have three English senses with the same lemma *bank* but from different resources, and two Italian senses from two different resources.

BableNet provides a mapping to WordNet 3.0, and since it covers all of WordNet, each of the 117K WordNet synsets is mapped to a single BabelNet multi-synset. However, since BabelNet has 15M multi-synsets, the majority of them are not covered by any WordNet synsets.

2.8.4 Homonym Lists

The following lists of homonymous words are accessible in the GitHub repository[11] of the current project.

[10]https://babelnet.org/stats

[11]https://github.com/AmirAhmadHabibi/ homonym_detector/tree/master/data/ homonym_lists

List of Type-A Homonyms

Hauer and Kondrak (2020a) have created a list of type-A homonyms extracted from dictionaries, including the English Oxford Living Dictionary and the Concise Oxford Dictionary of English Etymology. They call these homonyms type-A due to their etymological differences. This list consists of 804 lemmas and 1967 different etymologies, with each lemma having two to six distinct etymologies.

Wordsmyth Homonyms

As a part of a homonym annotation task, Rice et al. (2019) collected a list of 534 homonymous words form the Wordsmyth dictionary[12]. They consider each entry of a word in the Wordsmyth dictionary as a lexeme and annotate the homonymous words in a corpus with their corresponding Wordsmyth entry. Nevertheless, our use of this resource is only limited to the list of words and not their annotations.

Wikipedia Homonyms

This is a list of 258 homonymous words that was provided in Wikipedia until March 2020 under the name of "List of true homonyms".

Wikipedia Homographs

Wikipedia also included a list of 234 homographs until March 2020 under the title of "List of English homographs". Although this list is not a list of homonyms, we consider it among our homonym lists because of the great number of homonymous words that have overlap with homographs.

CrowdTruth Homonyms

This list of 247 homonymous words was obtained based on human judgment on the ambiguity of words in a corpus, as a part of the annotations in the work of van den Beukel and Aroyo (2018).

[12]https://www.wordsmyth.net/

Winnipeg Homonyms

This is a list of 933 homonymous words from the Upundo website[13], which was originally obtained from the website of the University of Winnipeg[14].

Google Homographs

This list of 162 homographs[15] was published by Gorman et al. (2018).

2.8.5 ODE Sense Clusters

Navigli (2006) presents an automatically built clustering of similar senses of words from WordNet, in order to mitigate the problem of fine granularity of WordNet senses. In this process, they used the lexemes in the Oxford Dictionary of English (ODE) and mapped WordNet senses to those lexemes using their definitions. Afterward, senses that were mapped to the same lexeme were clustered together. In this process, they created senses clusters for 16109 words, each having up to 31 clusters.

2.8.6 OntoNotes Sense Clusters

Hovy et al. (2006) manually create a clustering of similar senses of words in WordNet as a part of the OnetoNotes project[16]. Their resource includes 3785 words, each having up to 27 sense clusters. The senses are clustered together manually by a linguist based on their similarities to increase the inter-annotator agreement in the sense annotation of a corpus.

2.8.7 EuroSense Corpus

EuroSense is a sense annotated multilingual parallel corpus[17] presented by Delli Bovi et al. (2017). This corpus is a sense annotated version of the EuroParl, which is a parallel corpus of 21 languages. The sense annotation of EuroSense is done automatically using senses from BabelNet.

[13]http://www.opundo.com/homonyms.htm
[14]http://ion.uwinnipeg.ca/ clark/cog/norms/hom.html
[15]https://github.com/google/WikipediaHomographData
[16]https://catalog.ldc.upenn.edu/LDC2013T19
[17]http://lcl.uniroma1.it/eurosense/

Chapter 3

Prior Work

In this chapter, we first discuss the work on homonymy. There has not been a substantial amount of research done on the problem of distinguishing between homonymy and polysemy. However, there is related research done on homonymy in general, which can help in our task. Afterward, we discuss the work on sense clustering. The problem of the fine granularity of WordNet had motivated some research on clustering similar senses of sense inventories, which is related to our research. Finally, we discuss the research on detecting semantic relations between words through translation information which is another line of research pertinent to our task.

3.1 Homonymy

A recent attempt at distinguishing between homonymy and polysemy has been the work of van den Beukel and Aroyo (2018). They try to detect homonymous words in order to facilitate the task of humor recognition in short texts. However, they have a broader definition of homonyms than ours. Their definition includes both homographs, which are different concepts with the same word, and homophones, which are different words with the same pronunciation. Our focus was only on their work on homography, which corresponds to our definition of homonymy. Their method of homograph detection utilizes synset definitions from WordNet. In their paper, van den Beukel and Aroyo describe this method as keeping the synsets that have no overlap in their definitions and consequently considering any words with more than two remaining

definitions to be homographs. In other words, the method is to cluster the synsets of a word with similar definitions and consider words with more than two clusters to be homographs. Nonetheless, this description does not provide sufficient information and is not accurate compared with the implementation in the source code they provided[1]. The first difference is that the overlap of the definitions is not considered a hard constraint for the similarity of synsets. Instead, they compute a ratio of the overlapping words to all the words in the two definitions, excluding stop words (e.g., *and*, *he*, and *what*). Then they compare that ratio to the threshold of 0.1 to decide if two synsets are similar. The second difference is an additional similarity measure, which is WordNet's path similarity[2]. Any two synsets with the path similarity greater than or equal to the threshold of 0.3 are considered similar. Consequently, if two synsets are similar based on either of the above criteria, they would be considered related. The final difference is that the number of clusters has to be *more than one* for the word to be classified as a homograph. However, this number of clusters was stated as *more than two* in the paper, probably by mistake. They report an F_1 score of 49.5%, the precision of 35.3% and the recall of 82.5%, for the use of this method on their own list of 247 homonyms. Their list of homonyms is obtained manually based on the ambiguity of words in a corpus. Their results are confirmed by their provided implementation. We compare that implementation to our methods in Chapter 5.

Additionally, Beekhuizen et al. (2018) worked on an automatic method of distinguishing between monosemous, polysemous, and homonymous words, without considering their senses from sense inventories. Instead of using sense inventories, they used word embeddings to make a criterion for their classification by comparing them to their *context*. The context of a word is identified by its dictionary definitions or its context in a corpus or the semantic neighbors of the word in the vector space. They claim that the embedding vector of a monosemous word should be close to that of its context compared with poly-

[1] https://github.com/svenvdbeukel/Short-text-corpus-with-focus-on-humor-detection/

[2] A similarity score between 0 and 1, based on the shortest path in WordNet's hypernymy graph.

semous and homonymous words that should have a greater distance. Likewise, polysemous words should be closer to their context compared with homonymous words and their context. To test their hypothesis, Beekhuizen et al. combined the Wordsmyth[3] dictionary with some other resources and obtained 429 homonymous, 4672 polysemous, and 1229 monosemous words. Afterward, they computed the distance of these words to their context. Their experiments support their claims and demonstrate a clear distinction between the three classes of words in terms of the distance between their embedding vector and the embedding vector of their context when using dictionary definitions as the context. However, their results only demonstrate a general difference in the average distance for each class of words and they have not used their method to classify words into homonymous and polysemous. Despite the lack of focus on homonymy between *senses*, this work is an important step in the research on homonyms, demonstrating the potential for the use of word embeddings for homonym detection.

Another study on homonyms is the recent work of Rice et al. (2019). Their work is focused on finding estimates of the meaning frequency of homonyms. They obtained a list of 534 homonyms from the Wordsmyth dictionary. Then they manually annotated the use of these homonymous words in 50K lines of subtitles, along with 5K homonym-associate[4] pairs, with more than 90% agreement between the annotators. The use of Wordsmyth instead of WordNet for the annotations can be an essential factor in the high level of inter-annotator agreement, because, as Rice et al. mention, Wordsmyth has a coarser granularity of senses than WordNet. To be more precise, Wordsmyth has a two-level hierarchy of senses that first delineates between homonyms of a word and then makes a distinction between the polysemous senses. We analyze their list of homonyms using our homonym detection methods. Additionally, we use their list in building a new resource of type-B homonyms.

Finally, the work of Hauer and Kondrak (2020a) presents hypotheses about

[3]https://www.wordsmyth.net/

[4]The term *associate*, here, is used for words indicating the correct meaning of a homonymous word.

the characteristics of homonyms. They propose the "One Homonym Per Translation" (OHPT) hypothesis, which claims that each translation of a homonymous word can only represent one of its homonyms, or in other words, homonymous senses generally do not share any translations. They explain that since homonyms are semantically unrelated words that happen to have the same orthographic form, there is no reason to expect those two words to have any translations in common. Hauer and Kondrak propose similar hypotheses in the context of discourses, collocations, and sense clusters. To evaluate the OHPT hypothesis, they manually create a list of type-A homonyms and obtain translations from sense-annotated parallel corpora. The creation of the homonym list involved gathering all of the words that had senses with different etymological origins in English dictionaries such as the English Oxford Living Dictionary and the Concise Oxford Dictionary of English Etymology. This process led to the creation of a homonym resource that contains 804 lemmas corresponding to 1967 distinct etymologies. Further statistics of this resource are presented in Chapter 2. Having this resource, and using the translation of sense annotated corpora, they established that their OHPT hypothesis holds for more than 99% of the cases. The exceptions found to this hypothesis include the borrowing words and cognates, something we refer to as *parallel homonyms*, which is discussed in more detail in Chapter 2. The OHPT hypothesis is used as a basis of one of our primary methods to discover evidence for homonymy.

3.2 Sense Clustering

One of the attempts at solving the problem of WordNet's fine granularity is the work of Hovy et al. (2006) in the OntoNotes project. Unlike the work of van den Beukel and Aroyo (2018), Hovy et al. did not work on detecting homonymous words, but rather their focus was on clustering similar senses together. Their primary task was the manual annotation of a corpus of English and Chinese, and among their annotations were sense annotations. However, the fine-grained sense distinction of WordNet would have decreased the

agreement between the human annotators. Therefore, they created a manual clustering of WordNet senses, so that the very similar senses would be considered a single sense. Hovy et al. created a process in which after the sense annotation of each small sample by human annotators, they gave a score to the annotations based on the agreement between the annotators, and a score below 90% would have led to a revision of the sense groupings by a linguist. This amount of manual labor means a high accuracy. However, manual work is more time consuming, which limits the coverage of the vocabulary. Their sense clustering includes only 3.8K words out of the 26K words that have more than one sense in WordNet. Nevertheless, we use this resource as a part of the process of building our gold standard data in Chapter 5. Further details of this resource are presented in Chapter 2.

Another work on clustering WordNet senses is that of Navigli (2006), published in the same year as the work of Hovy et al. (2006), with an essential difference of avoiding the manual effort. Navigli proposed an *automatic* method of creating clusters of WordNet senses. Their method required a mapping from the WordNet senses to the entries in the Oxford Dictionary of English (ODE). ODE provides a hierarchy of senses, which can distinguish between homonymy and polysemy. Therefore, having a mapping from WordNet senses to ODE entries would have provided a reliable grouping of senses of each word. Navigli used lexical and semantic methods to map senses from WordNet to entries from ODE through the similarities between the definitions provided by the two resources. His lexical method for matching senses of WordNet to senses of ODE simply used the overlap between the lemma in the two definitions as a measure of similarity. However, in some cases, the definitions did not share any lemmas. Therefore, he proposed an additional semantic method, an algorithm that exploited WordNet relations and collocation information of words to detect similarity between the lemmas in the two definitions, and using that, it measured the semantic relatedness between the definitions of senses. To evaluate his method, he created a gold-standard data of mapping WordNet senses of 763 words to their respective ODE entries. He reports an F_1 score of 83% for his best method, which is the semantic method. Navigli's automatic

process makes it easier to have a wide-coverage of sense clustering, and because of that, we use the resource he provided in our work. On the other hand, automatic clustering is usually not as reliable as the manual clustering and, as expected, we have found many cases of incorrect clusters in this resource, which are discussed in Chapter 5.

3.3 Finding Semantic Relations

The work mentioned in the previous section was focused on connecting semantically related senses using their *dictionary definitions*, either automatically or manually, through human interpretation. However, *dictionary definitions* are not the only sources of information for this task. Another line of research has tried detecting semantic relations, such as synonymy between words and polysemy-homonymy between senses, from translation or co-occurrence information.

One of the attempts at extracting semantic relations from translations was the work of Dyvik (2004, 2009). He worked on theories to build a sense inventory out of the relations extracted from parallel corpora of text. He introduced an algorithm called semantic mirroring, which can partition translations of a word type into clusters that would represent its senses. In other words, the algorithm induces the senses of a word from its translations; therefore, it does not require any sense inventories. Dyvik presents promising examples of the use of his algorithm in his paper. However, he does not provide any evaluation of the performance of this sense induction algorithm on a larger scale. The sense induction algorithms are not directly involved in our task of detecting homonymy. However, the same techniques that are used to find semantic relations between the tokens of words can be used in detecting semantic relations between the senses of a word, which would then be directly useful for our task. Therefore, some of our methods are inspired by these algorithms. In particular, Dyvik's semantic mirroring is fundamental to two of our theorems, and so we will leave the details of this algorithm to be discussed in Chapter 4.

Another research that utilizes translation information in finding semantic

relations is the work of Apidianaki (2008) and Bansal et al. (2012) . Similar to the work of Dyvik (2009), the main focus of these projects is inducing senses of a word from its translations in a parallel corpus. However, the difference is that they build feature vectors for translations of a word and cluster the translation based on these vectors. These feature vectors are built using the contextual information from the source side of the parallel corpus. The clusters of translations created based on these vectors are then considered to be representatives of the senses of the initial word.

Translation has proved beneficial in detecting semantic relatedness in other research as well. Van der Plas and Tiedemann (2006) worked on obtaining semantic relations between words from distributional information in monolingual and multilingual data. In other words, they have tried to detect synonymous words by similarity in monolingual context and in translation. They have demonstrated a 7% improvement in F_1 score when using translations over the use of distributional information in only monolingual data. This improvement motivates us to focus on the use of translation information in detecting semantically related senses of words.

Chapter 4

Methods

This chapter describes our methods of distinguishing between homonymy and polysemy, in two main sections: First on the word level and then on the sense level. The first section describes our method of classifying a word into polysemous or homonymous, using a graph of concepts, where each concept corresponds to exactly one sense of the word. The second section is about finding semantic relations between the concepts of a word. These semantic relations constitute the edges of the word-level graph described in the first section.

4.1 Detecting Homonymous Words

Our homonym detection method builds a graph for each word. This graph has concepts that are represented by that word as its nodes, and the edges of this graph indicate semantic relations between the concepts. The semantic relatedness between each pair of concepts is determined by the methods explained in the next section. We first determine the semantic relatedness of each pair of concepts corresponding to the senses of the word in question, and if related, an edge is inserted between the pair of concepts. Afterward, if the senses of the word compose more than one connected component, the word is classified as homonymous and otherwise as polysemous. Figure 4.1 demonstrates an example of such a graph, where the senses of the word $mint_n$ form two connected components, indicating homonymy.

Considering a connected component as a lexeme also means that two senses

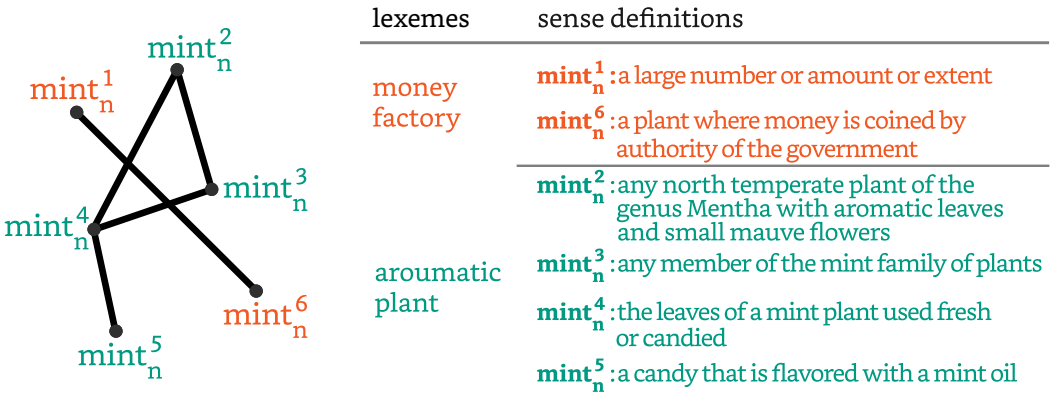

Figure 4.1: An example of a graph of semantic relations.

might be in the same lexeme but not directly connected. In other words, we are considering the transitivity of semantic relatedness within the concepts of a word. For example, in Figure 4.1, even though there is no direct edge between $mint^5_n$ and $mint^3_n$ to indicate their semantic relatedness, these two senses are indirectly connected by a path that goes through $mint^4_n$. The transitivity assumption means that missing an edge in the graph would not always change the result of our classification of a word.

4.2 Finding Semantic Relations Between Senses

We present five methods to detect semantic relations between senses and discuss an algorithm by Dyvik (2004), which is fundamental to two of our methods. All of our methods, except for the last one, utilize translation information to find semantic relations between WordNet synsets. These synsets represent senses of a word, and if two synsets are semantically related, then the relation between the senses of a word in those two synsets is that of polysemy.

4.2.1 One Homonym Per Translation

Our first method is based on the One Homonym Per Translation (OHPT) hypothesis of Hauer and Kondrak (2020a), which states that different lexemes of a word do not share any translations. We discussed this hypothesis along with its corollary and the exception of parallel homonymy in Chapter 2. Consid-

ering these exceptions, we rephrase Corollary 1.1 of OHPT hypothesis in the following theorem:

Theorem 1. *If two senses W^i and W^j of a word W in one language are translated by the word Z in another language, then one and only one of the followings holds:*

- *W^i and W^j are polysemous.*
- *W and Z are parallel homonyms.*

We know that parallel homonymy is a rare phenomenon. Therefore, using Theorem 1, our first method considers shared translations as evidence indicating polysemy of senses. For example, consider the English word $budget_n$ with the following senses in WordNet:

1. $budget_n^1$: *A sum of money allocated for a particular purpose.*
2. $budget_n^2$: *A summary of intended expenditures along with proposals for how to meet them.*

Both of these senses are translated in BabelNet with the Spanish word *presupuesto*. Therefore, based on Theorem 1, it is either the case that the two senses of *budget* are semantically related and therefore polysemous, or *budget* and *presupuesto* are parallel homonyms. Our method simply rejects the second case and accepts the first one. Therefore, $budget_n^1$ and $budget_n^2$ will have the relation of polysemy.

4.2.2 The Issue of Parallel Homonyms

We mentioned that we reject the assumption of parallel homonymy in the use of Theorem 1, as this is a rare phenomenon. However, we can also try to automatically detect some of these rare cases, in order to diminish their effect even more.

In order to detect parallel homonyms, we rely on orthographic similarity of words, hoping that it would make us able to detect both cognates across languages and spelling variations within a language. Our method computes the edit distance (Levenshtein distance), which is the minimum number of character insertions, deletions, and substitutions, between the two words. This

number is then normalized by a division to the length of the longer word. If this value exceeds a certain threshold, our method considers those two words as orthographically similar, and consequently, cognates or spelling variations. Of course, this method can classify many similar words as cognates, that might not have the same etymology. However, considering that we are only checking those words that are suspected to have a semantic relation, it becomes less likely for this method to incorrectly classify words as cognates.

4.2.3 Dyvik's Algorithm

Before presenting our next semantic relation method, we discuss an algorithm that is fundamental to our next two methods. Dyvik (2004, 2009) has worked on theories to automatically build a sense inventory from the relations extracted from parallel corpora of text. His main assumption was that an overlap between the translations of two words indicates semantic relation between them. He then proposed to induce senses of words using a process that he calls "semantic mirroring". This process is illustrated in Figure 4.2 using Dyvik's own example. His algorithm involves two steps of semantic mirroring. In this example, the first step is from the initial Norwegian word *tak* to its English translations in what he calls the first t-image. The second semantic mirroring is from the words in the first t-image to their Norwegian translations in what is called the inverse t-image. After these two steps, a clustering of the English translations of the first t-image can be obtained using their shared translations in the inverse t-image. In this process, the initial word *tak* is removed from the inverse t-image. Therefore, the first t-image is partitioned into three clusters: (ceiling, roof), (cover), (hold, grip). These three clusters correspond to three senses of *tak*. Therefore, through this process, Dyvik could distinguish the senses of a word.

Algorithm 1 presents our pseudocode implementation of his method. The inputs of the function are the initial word and mappings between words to their translations in both directions. First, the function creates a list of the translations of the initial word in $firstImg$, which is the first t-image. Afterward, the clusters of translations are initialized, putting each word of the first

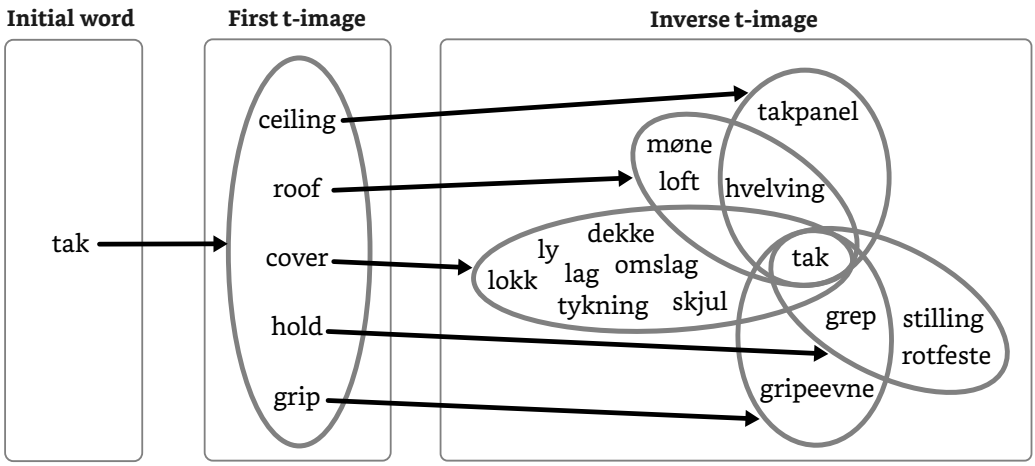

Figure 4.2: An example of the semantic mirroring based on Dyvik (2004), where the translations in the first t-image will be clustered together if they share translations in the inverse t-image, except the initial word.

t-image in a singleton cluster. The *while* loop then merges related clusters until there are no clusters that are related. In each iteration of the *while* loop there is a *for* loop that goes through all pairs of clusters. For each pair of clusters then the second *for* loop determines if they should be merged. This merge would happen if a pair of words from two clusters is found having a shared translation in the inverse t-image, except for the initial word as it is removed from the inverse t-image. Finally, the algorithm returns the merged clusters.

4.2.4 Two Words, Two Translations

Our work on Dyvik's algorithm and its implications leads to more theorems. Going back to the example in Figure 4.2, we can observe that the necessary condition for clustering any two words of the first t-image together is that they should share two translations in the inverse t-image, one of which is the original word *tak*. For example, the two English words *ceiling* and *roof* are clustered together because they share two Norwegian translations *tak* and *hvelving*. Therefore, any two English words need at least two Norwegian translations to be considered semantically related. Furthermore, Dyvik's clustering

Algorithm 1 Dyvik's translation clustering

Inputs:
$inWrd$: Input word to analyze
s2t(w) : Translation function from source language to target language
t2s(w) : Translation function from target language to source language
function DyvikAlg($inWrd$, s2t, t2s)
    ▷ create $firstImg$ as a set of words
    $firstImg \leftarrow$ s2t($inWrd$)
    ▷ Initializing the clusters of $firstImg$
    $clusters \leftarrow \{\}$
    for $tWrd \in firstImg$ **do**
        add $\{sWrdSet\}$ to $clusters$
    ▷ merge the clusters based on the inverse t-image
    while there are clusters to merge **do**
        for $c1, c2 \in clusters$ **do**
            for $w1 \in c1$ and $w2 \in c2$ **do**
                if t2s($w1$) $\cap$ t2s($w2$) $-\{inWrd\} \neq \emptyset$ **then**
                    merge $c1$ and $c2$
    return $clusters$

is limited to the word types, yet, it can be extended to those senses of the words that are involved in the translations, which would provide a clustering of senses, instead of words. This extension of Dyvik's algorithm to senses of words, in addition to the earlier observation, eventually, led to the development of Theorem 2 (Kondrak et al., 2020).

Theorem 2 (Two Words, Two Translation)**.** *If two distinct words X and Y in one language both translate into two different words W and Z in another language, then one and only one of the followings is true:*

- *All senses involved in all those translation instances are semantically related.*
- *At least two of the four words are homonymous.*

This theorem is stricter than Theorem 1 in that it requires more evidence to suggest a semantic relation. The previous Theorem 1 requires two words to indicate a semantic relation. These two words are a source word and a shared translation between senses of that source word. However, Theorem 2 requires

four words in order to infer a semantic relation. These four words are two shared translations between senses of two words.

As an example of the application of this theorem, we take the two English words $graze_v$ and $pasture_v$ that are both translated with the Spanish words $apacentar_v$ and $pacer_v$. Based on Theorem 2, and due to the rareness of parallel homonymy, our method is to consider any senses involved in these translations to be semantically related. For instance, among the senses involved in these translations are $graze_v^1$ translated to $apacentar_v^2$, and $graze_v^3$ translated to $pacer_v^1$. Therefore, $graze_v^1$ and $graze_v^3$ are semantically related, and our method classifies the relation between these two senses as polysemy.

4.2.5 Two Synsets, Two Words

If we take the clusters created by Dyvik's algorithm in the sense level and consider any two senses in the same cluster to be semantically related, we would observe that Theorem 2 will not cover all of these semantic relations. An example of such cases is when two senses of the word W in the first t-image are both translated by the word X in the inverse t-image. In this case, the two senses of W will be clustered together, yet Theorem 2 does not cluster such cases. The integral theorem is Theorem 3, that combined with Theorem 2 can replicate the result of Dyvik's algorithm in clustering senses.

Theorem 3 (Two Synsets, Two Words)**.** *If two distinct words X and Y in any two languages share two multi-synsets $Syn1$ and $Syn2$, then one and only one of the followings is true:*

- *$Syn1$ and $Syn2$ are semantically related.*
- *X and Y are both homonymous.*

Theorem 3 is a generalization of Theorem 1, stated in terms of multi-synsets in multilingual sense inventories. Here sharing a multi-synset for two words means having senses that translate each other. Moreover, in Theorem 1, the two words needed to be from different languages, yet Theorem 3 requires any two words with no restriction on the languages being different. For example, consider these two concepts from BabelNet:

- *bn:00026603n* : *An event that will inevitably happen in the future.*
- *bn:00026604n* : *The ultimate agency regarded as predetermining the course of events.*

Both of these multi-synsets include senses of the English words *fate* and *destiny*. Therefore, our method based on Theorem 3 considers these two multi-synsets to be semantically related since the parallel homonymy of the two English words is unlikely. As a result, the two senses of *fate* that correspond to these two multi-synsets would be polysemous, and the same is true for the two senses of *destiny*. On the other hand, Theorem 1 required senses in a second language to conclude these semantic relations.

4.2.6 Three Multi-synsets

Another theorem we worked on is Theorem 4 (Kondrak et al., 2020), which is stricter than our previous theorems as it would require three related words to assume semantic relatedness.

Theorem 4 (Three Multi-synsets)**.** *If three pairs of senses* (X^1, Y^1), (X^2, Z^2), *and* (Y^3, Z^3) *of the different words* X, Y, *and* Z *are pairwise synonymous, then one and only one of the followings is true:*

- *All those senses are semantically related.*
- *At least two of the words are homonymous.*

As an example of the use of this theorem, we can mention these three multi-synsets in BabelNet:

- *bn*:00082755*v* : *Have an argument about something.*
  - Senses: EN:$debate_v^4$, ES:$discutir_v^5$
- *bn*:00085651*v* : *Think about carefully; weigh.*
  - Senses: EN:$deliberate_v^1$, EN:$debate_v^2$
- *bn*:00086342*v* : *Discuss the pros and cons of an issue.*
  - Senses: ES:$discutir_v^2$, EN:$deliberate_v^2$

The Spanish word *discutir* and the two English words *debate* and *deliberate* share these three multi-synsets. Based on Theorem 4, and with the assumption

of the rareness of parallel homonymy, our method considers the three multi-synsets in the above example to be semantically related. Consequently, those senses of each word that are involved in the above multi-synsets, such as $debate_v^4$ and $debate_v^2$, are considered polysemous to each other.

4.2.7 Hypernymy & Hyponymy

All of our methods, so far, utilized translation information to infer semantic relations. Nonetheless, wordnets provide us with some other kinds of semantic relations between synsets that can lead to polysemy relations. These are hypernymy-hyponymy and meronymy-holonymy.

Here is a reminder of these relations:

$$\underbrace{X}_{hyponym} \; is\ a \; \underbrace{Y}_{hypernym}$$

$$\underbrace{X}_{meronym} \; is\ part\ or\ member\ of \; \underbrace{Y}_{holonym}$$

Our hypothesis is that in the graph of hypernym-hyponyms and meronym-holonyms, the sister nodes are semantically related. We consider the nodes having the same hypernym or holonym as sister nodes.

Then within the senses of each word, our method considers all sister nodes to be polysemous. For example, the English word *match* has, among others, these two synsets:

- *bn*:00036522*n* : *Lighter consisting of a thin piece of wood or cardboard tipped with combustible chemical; ignites with friction.*
  - Senses: $match_n^1$
- *bn*:00053773*n* : *A burning piece of wood or cardboard.*
  - Senses: $match_n^3$

Both of these synsets are hyponyms of this synset:

- *bn*:00045882*n* : *A device for lighting or igniting fuel or charges or fires.*

Therefore, our method considers $match_n^1$ and $match_n^3$ to be polysemous to each other.

We do not expect a high recall from this method but a high precision. Because, intuitively, many semantic relations cannot be retrieved through a

shared hypernym or holonym, but a shared hypernym or holonym can indicate semantic relatedness.

Chapter 5

Experiments

This chapter first discusses our data collection process, which includes extracting translation for senses and obtaining a gold dataset of homonymous and polysemous words. In addition, we discuss building and updating two homonym resources. Afterward, we explain the use of our homonym detection methods and the translations extracted in the previous step. We start by using the translations of single languages and then we combine the translation data of different languages. Next, we discuss our experiments on the best combination of languages and their results. Eventually, we analyze the errors.

5.1 Data Collection

Our experiments are on detecting semantic relatedness of different senses of words through their translation. Therefore, a natural first step is to collect translations for the English senses. These translations are obtainable from two categories of sources: parallel corpora and sense inventories. In the following sections, we discuss that our attempts to obtain high quality and high coverage translations from parallel corpora did not succeed. However, we could eventually achieve that from sense inventories.

5.1.1 Obtaining Translation from Parallel Corpora

Translation of words can be obtained from parallel corpora of text, which provide sentence by sentence translations. The way to achieve this is through a process of automatic word-alignment, which uses distributional information

of words and their co-occurrence on the two sides of a parallel corpus to map the words of each sentence to their translations. To run this process, we use fast_align[1], which is an unsupervised word aligner (Dyer et al., 2013).

Our experiments require translations of *senses* of words. However, the above process would provide us with translations for word *types* in a corpus. Therefore, we require our parallel corpus to be sense annotated, on the English side.

We began our experiments with EuroSense (Delli Bovi et al., 2017), which is a multilingual sense annotated parallel corpus of 21 languages. The word-alignment of this corpus provided us with a mapping of word senses to their translations. Nevertheless, these experiments were not successful for three primary reasons:

1. The accuracy of the automatic word-alignment process was not sufficient to produce a reliable mapping between words and their translations.
2. The sense annotations of EuroSense were not manual, and this automatic process caused incorrect annotations.
3. The coverage of EuroSense was limited for many homonyms. That is explainable by the rareness of many senses of these words and the particular nature of the EuroSense corpus, which comes from the proceedings of the European parliament.

We believe that the first two reasons, put together, could aggravate the problem of incorrect translations. The absent senses in the corpus, in addition to the incorrect translations, made our goals even less achievable through this resource. As an example of these problems we can mention the two following sentences from EuroSense: "En: We have many common tasks." and "It: Ci attendono molti compiti comuni.". The sentences are not literal translations of each other and even though "many common tasks" means "molti compiti comuni", the literal translation of "Ci attendono" is "there awaits" and not "We have". When the translation is not literal, the error rate of the word-aligner will increase as it is trying to map a word to a translation that does not exist in the target sentence. Moreover, even when the translations are

[1]https://github.com/clab/fast_align

not incorrect they can be inexact and involve hypernyms and hyponyms. For example, "apple" might be translated to a word that means "fruit". The errors of the word-alignment process are not restricted to the two mentioned cases. These problems encouraged us to use sense inventories instead. Sense inventories have greater coverage of senses since they are usually independent of any topic. In addition, they do not require us to perform automatic word-alignment and risk adding to the noise in translations.

5.1.2 Obtaining Translation from Sense Inventories

We begin this part of our experiments by using MultiWordNet, which is an English-Italian sense inventory. Afterward, we extend our experiments with BabelNet, which is a larger multilingual wordnet, including more than 280 languages. Additional information and statistics about both of these resources are provided in Chapter 2. These resources provide us with multilingual synsets, which means access to the translations of different senses of English words in different languages. Nevertheless, these sense inventories are not without deficiencies. Similar to the use of parallel corpora, problems of insufficient or incorrect translation appear here as well, yet, on a smaller scale.

Sparsity Problem

The first problem is the sparsity of synsets. This problem is particularly discernible in the cases of the infrequent concepts, especially when limiting the translations to only one language. For example, consider the English word $debt_n$. This word has, among others, the following two synsets in BabelNet (Each synset is represented here by its synset ID and definition in the first line, followed by the senses in different languages in the next lines):

1. *bn*:00025638*n* : *Money or goods or services owed.*
    - English: *debt*
    - Italian: *debito*, *debiti*, *chiodo*
    - Spanish: *deuda*, *deudas*
2. *bn*:00025639*n* : *An obligation to pay or do something.*
    - English: *debt*

- Italian: -
- Spanish: *deuda*

We can judge by the definitions that these two synsets should be semantically related. Nevertheless, if we only consider Italian translations, the second sense of $debt_n$ would be left with no translations and then there would be no evidence of semantic relatedness between these two senses.

A solution to this problem would be to take more languages into consideration and determine semantic relatedness by checking the translations of these synsets from more than only one language. In this example, adding Spanish translations would provide the Spanish word *deuda*, which is a shared translation between both of the synsets. This shared translation would be evidence for the semantic relatedness of the two senses of $debt_n$.

Incorrect Translations Problem

The second problem is that of the incorrect translations. In particular, BabelNet is prone to have incorrect translations because it is automatically built through the integration of 47 resources such as Wikipedia, WordNet, Wiktionary, etc. Moreover, BabelNet uses resources such as the machine translation of WordNet, which in itself can have incorrect translations, due to deficiencies in current methods of machine translation.

We try to reduce the number of incorrect translations by excluding those resources that are of lower quality. We are able to analyze the quality of the resources since each sense in BabelNet has its original resource indicated. Our analysis led us to the conclusion that some of these resources contain more incorrect translations compared to the others. These resources included, among others, machine translations of WordNet and Wikipedia. Therefore, as the first step to mitigate the problem of incorrect translations, we tried excluding senses belonging to these resources of lower quality. However, contrary to our expectations, excluding these resources did not improve the results. Therefore, we can say that in general, the positive effect of the correct translations of the low-quality resources overcomes the negative effect of their incorrect translations.

Additional Restrictions

In order to prepare the data of the sense inventories for our experiments, we add two more restrictions.

First, we limit the synsets to those that are only present in Princeton WordNet. That is because our gold standard data (discussed in the next section), covers only WordNet senses. Therefore, to be able to compare any results to our gold data, we would need the test data to have the same set of synsets. For example, the English word $problem_n$ has, among others, these three synsets in BabelNet:

1. *bn*:00048242*n* : *A state of difficulty that needs to be resolved.*
    - English: *problem*, *job*
    - Italian: *problema*
    - Spanish: *problema*, *dificultad*, *lío*, *Problem*
2. *bn*:00064526*n* : *A question raised for consideration or solution.*
    - English: *problem*
    - Italian: *problema*
    - Spanish: *problema*, *ejercicio*
3. *bn*:01917408*n* : *A mathematical problem is a problem that is amenable to being represented, analyzed, and possibly solved, with the methods of mathematics.*
    - English: *Mathematical problem*, *Mathematics problem*, *Problem*
    - Italian: *problema matematico*
    - Spanish: *problema matemático*

The first two synsets are covered by WordNet. The synset *bn*:00048242*n* is mapped to $problem_n^1$ in WordNet, and *bn*:00064526*n* is mapped to WordNet's $problem_n^2$. However, the third synset is originated in Wikipedia and is not covered by WordNet, so it cannot be mapped to any senses in our gold data. That means if $problem_n$ is in our test data, we would not be able to evaluate the third sense.

Another restriction is to include only single-word senses in the synsets. In other words, any senses that are compound words or have extra information

in parenthesis, are removed. For example, we would only consider the first of the following French senses: *rayon*, *rayon_(optique)*, and *rayon_lumineux*. This restriction slightly enhances the results of our experiments, which are discussed later in this chapter. The slight improvement caused by this restriction can be interpreted as the higher amount of incorrect translations among the compound senses. This interpretation is also strengthened by our observation of the Persian translations of BabelNet.

5.1.3 Creating the Gold Data

In order to evaluate our methods, we require lists of positive and negative cases of homonymy. Our positive cases belong to a manually built list of homonyms, discussed in the next section. The negative cases, on the other hand, can be any words with multiple senses that are not in our homonym list. This is because homonymy is a rare phenomenon, and we can assume that our homonym list includes the majority of English homonyms. Therefore, the rest of the words in the English vocabulary are considered to be either monosemes or polysemes. However, we take further steps to make sure of the validity of our polyseme list, which is described in section 5.1.3. Moreover, our experiments are based on WordNet senses and our methods should be limited to the senses that are present in WordNet. Therefore, homonymous words that have only one of their lexemes covered by WordNet senses are not considered homonymous in our experiments.

Type-A Homonyms

This list of homonyms was originally built by Hauer and Kondrak (2020a), using English dictionaries. The list consists of 804 lemmas of type-A homonyms, which are homonyms with distinct etymologies. These 804 lemmas correspond to 1957 homonyms from different etymologies.

Along with this list, Hauer and Kondrak (2020a) have provided a mapping of 5203 WordNet senses associated with this list of homonyms. An instance of this mapping is illustrated in Table 5.1. In this table, we can observe that several WordNet senses can be mapped to one homonym; however, each

Lemma	Lexeme number	POS	Synonym	WordNet sense	WordNet definition
chord	*100*	*n,v*	*harmony*	$chord_n^2$	*combination of musical notes*
				$chord_v^1$	*play chords on instrument*
				$chord_v^2$	*bring into harmony in music*
chord	*200*	*n*	*line*	$chord_n^1$	*straight line connecting points*

Table 5.1: A snippet of the mappings from the type-A homonym list to WordNet senses.

WordNet sense can be mapped to at most one homonym.

The homonym to WordNet sense mappings had problems primarily related to the use of the ODE clustering (Navigli, 2006) in the process of building the resource. In this process, considering the fine granularity of WordNet senses, the type-A homonyms were mapped to ODE sense clusters instead of the single senses, to accelerate the manual mapping. However, the ODE sense clusters were automatically built and had missing senses and incorrect clusters. The missing senses in ODE clustering, lead to missing senses in the homonym mapping, and the incorrect sense clusters, created incorrect mappings between senses and homonyms.

We tried to mitigate these problems manually. First, we found 580 WordNet senses missing from the ODE clustering of the 804 lemmas. We manually mapped these 580 senses to their corresponding homonyms. During this process, 46 senses were mapped to homonyms that did not match their part of speech (POS). For example, the homonym "*cricket.200*" referring to the game of cricket, has only *noun* in its list of POS; however, we had the WordNet sense $cricket_v^1$ meaning "to play cricket", which clearly should be mapped to that homonym. The second problem emerged from incorrect clusters, when all of the senses in that cluster were mapped to a homonym based on only one of the senses. Discovering all of these incorrect clusters would have required considerable manual work that was not possible in our limited time frame. Nevertheless, there were 21 cases that could have been detected automatically, as discussed by Hauer and Kondrak (2020a), using their One Homonym Per Sense Cluster (OHPSC) hypothesis and the high-quality manual sense clusters of the OntoNotes project (Hovy et al., 2006). These were among the cases

	Lemmas	Words(lemma+POS)	lexemes	Mapped senses
Original	804	1601	1957	5203
Updated	804	1603	1958	5783

Table 5.2: Statistics of the type-A homonym resource, before and after the update.

of the OntoNotes clusters combining senses mapped to different homonyms. We manually corrected these cases by breaking those incorrect clusters and individually mapping each sense to its correct homonym. The corrections and additions of mappings together amount to more than 600 sense mappings, which corresponds to more than 400 lemmas out of the 804. In addition, we have found one homonym "*second.100*" that was used in the mapping but was missing from the list of homonyms. We added this homonym to the homonym list. Table 5.2 summarizes the statistics of the updated resource. The updated resource is a part of the contributions of this thesis[2], which can facilitate future research on homonyms and the semantic relations between their senses.

After updating the type-A homonym resource, we require two additional changes in order to make this resource suitable for our experiments. The first change is related to POS of the words and the second is related to homonyms that are not present in WordNet.

The first change is due to our specific definition of homonymy, which requires different parts of speech of a lemma to be considered as different words. Based on this definition, in the example presented in Table 5.1, we consider $chord_n$ and $chord_v$ to be two different words. As a result, $chord_v$ can no longer be considered homonymous, since it only carries the meaning of "harmony" in our data. However, $chord_n$ is homonymous, since it covers two distinct meanings. Following the analysis of this example, we found that the 804 lemmas in the resource would constitute 1603 words with distinct POS, which include 888 homonymous words[3] and 715 non-homonyms.

Our second change is to restrict the homonyms to only those that are

[2]The link to the type-A resource: https://github.com/AmirAhmadHabibi/homonym_detector/tree/master/data/HK.1.2/

[3]We refer to a lemma with a distinct POS as a word.

present in WordNet. For example, the English word $scrip_n$ has three homonyms, the first one of which has a sense in WordNet (i.e. $scrip_n^1$):

1. *scrip.100* : A provisional certificate of money subscribed to a bank or company, entitling the holder to a formal certificate and dividends.
2. *scrip.200* : A small bag or pouch, typically one carried by a pilgrim, shepherd, or beggar.
3. *scrip.300* : Another term for script; story.

We do not consider words like $scrip_n$ in our list of homonyms since we are working on WordNet synsets and we require the word to be homonymous in that context. Excluding the homonyms that are not present in WordNet would leave us with 508 homonymous words (lemma+POS) corresponding to 1109 lexemes and mapped to 2511 WordNet senses.

Polysemes

We build our list of polysemes using the OntoNotes sense clustering (Hovy et al., 2006). This resource provides manually built clusters of WordNet senses for 4358 words. However, some words do not have all of their WordNet senses present in their clusters. We found 767 senses missing and we added each one in an individual singleton cluster for its corresponding word.

We do not use all of the 4358 words that have sense clusters in the OntoNotes resource. Instead, we select the words with more than one cluster, which amounts to 3232 words. This selection is due to our desire to experiment on words that are suspicious of homonymy, and based on One Homonym Per Sense Cluster hypothesis of Hauer and Kondrak (2020a), if a word has all of its senses clustered together, it is not a homonym. Therefore we eliminate the cases of clear polysemous words, and by eliminating these words, we add to the challenge of our task.

Additionally, to ensure that no homonyms are in this list, we remove any words that have their lemmas in the following lists of homonyms: the list of type-B homonyms discussed in the next section, type-A homonyms of Hauer and Kondrak (2020a), CrowdTruth homonyms of van den Beukel and Aroyo (2018), the homonym list of Gorman et al. (2018) The list of the University

of Winnipeg, Wikipedia list of homographs, and Wikipedia list of homonyms. A further description of these lists is in Chapter 2. Eventually, excluding the words that were in these lists, further reduces the number of our polysemes to 2137 words.

Creating A Balanced Dataset

Using the above resources, we create a balanced dataset of homonyms and polysemes. However, any natural sample of words would not be balanced because most of the words in any language are not homonyms. Our preference towards a balanced dataset is because of the nature of our task, which is asking about the classification of words into homonym or polyseme, when in *doubt*. We do not expect to receive as an input, those majority of the words that are definite non-homonyms. Therefore, we assume a 50% chance for both classes.

To build our balanced dataset, we remove the synsets that belong to named-entities. This is because of our definition of homonymy, which only considers senses representing concept and not proper names. We consider any synset that contains at least one proper name (capitalized lemma) in their WordNet lemmas to be a named-entity. As a result of this process, the number of senses or clusters for some words changes, since they might no longer fit in our previous definitions of homonymy or polysemy. These are the words that are left with less than two lexemes after the removal of the named-entities. Having these cases filtered out from our lists, we acquire a list of 474 homonymous words. We then randomly choose 474 words from our list of polysemes. Finally, we make a 20%-80% split of this data for test and development sets.

Creating the type-B Homonym Resource

We created a new homonym resource using a list that was a part of the homonyms used in the annotations of Rice et al. (2019). In their project, they annotated a corpus using a list of 534 homonymous lemmas that they acquired from the Wordsmyth dictionary. However, their homonyms were not mapped to WordNet senses and a mapping to WordNet was necessary for our experiments. Therefore, we had to create a manual mapping between these

homonyms and their corresponding WordNet synsets. First, we excluded those lemmas that were already in our type-A list, since their mapping to WordNet sense was available. These cases comprised 368 lemmas out of the entire list of 534, and by removing them, we reached a list of 166 new homonymous lemmas. Afterward, we searched for these lemmas in Wordsmyth online dictionary[4], and for each lemma, we created a list of their entries in Wordsmyth, which would be considered as different homonyms of that lemma. Moreover, we paired each entry with a synonym or a short description of its meaning for disambiguation, similar to the type-A list. This process provided us with 334 distinct homonyms. Having the list of homonyms and their synonyms, we manually mapped WordNet senses to their corresponding homonyms. To accelerate this manual process, we mapped the ODE sense clusters of WordNet (Navigli, 2006), instead of performing the mapping individually for each sense. During this process, we discovered more than 30 incorrect sense clusters in the ODE clusters, which contained senses from different homonyms. We broke these clusters down and mapped their senses individually. Additionally, there were 57 WordNet senses missing from the ODE sense clusters that had to be added. Furthermore, 97 Wordsmyth homonyms did not match any WordNet senses. Eventually, we created a list of 166 lemmas, corresponding to 295 words (i.e. lemma+POS). This list contains 334 homonyms, mapped to 740 WordNet senses[5].

Similar to the type-A homonym list, we had to make changes to this list in order to make it suitable for our experiments. First, we had to separate the lemmas by their POS and only consider the words that were homonymous. Second, we exclude any lexemes that were left without a mapping to WordNet senses. Eventually, we reached a list of 94 words that had their homonymous senses in WordNet. These words correspond to 201 lexemes, which were mapped to 257 WordNet senses. This final list of 94 words was used in our experiments, which are described in the following sections.

[4]https://www.wordsmyth.net/

[5]The link to the type-A resource: https://github.com/AmirAhmadHabibi/homonym_detector/tree/master/data/WSM1.0/

Since we assume the first list of homonyms of Hauer and Kondrak (2020a) covers almost all type-A homonyms, we consider this second list as a representative of type-B homonyms, unless we discover cases that we have evidence against it. Here are some of the cases of type-A homonyms that we discovered in the Wordsmyth list:

- "vet" has two lexemes related to "veterinary" and "veteran".
- "tower" is a homograph[6] with the meanings of "tall building" and "one who tows".
- "pod" has two lexemes related to "vessel" and "herd".
- "pat" has two lexemes with the meanings of "tap" and "perfect".
- "lighten" is a different form of the homonymous word "light", which is already in the type-A list, yet "lighten" is only in the Wordsmyth list.

We discovered these cases by finding their different etymologies through the Oxford Dictionary in https://www.lexico.com/. The list of type-B homonyms and their mapping to WordNet senses are another contribution of this thesis.

5.2 Development

We create graphs of semantic relations where the vertices are synsets and the edges are obtained through the various methods described in Chapter 4. Having such a graph of synsets for a word, we can determine whether or not it is homonymous, merely through counting the number of connected components of the graph.

Our experiments started with Italian translations from MultiWordNet. We then extended the translations to the Italian translations of BabelNet. Afterward, we worked on optimizing our models on all available languages in BabelNet. We found the combination of Indonesian and Spanish to be producing the best results. Finally, we tried our different theorems on the translations from the two best languages. These experiments are discussed in the following sections.

[6]Homographs are homonyms that their lexemes have different pronunciations.

English word	Italian word	Meaning 1	Meaning 2
band	banda	ring	group
bank	banco	ridge	repository
bongo	bongo	drums	antelope
canon	canonico	law	clergyman
colon	colon	currency	intestine
crane	gru	machine	bird
diet	dieta	food	assembly
mandarin	mandarino	language	citrus
palm	palma	tree	hand
port	porto	harbor	wine

Table 5.3: Parallel homonyms between English and Italian retrieved using MultiWordNet translations of the type-A homonym list.

5.2.1 Using OHPT

We begin our experiments using Theorem 1, which is a corollary of One Homonym Per Translation (OHPT) hypothesis. Our initial experiments are focused on discovering the exceptions to the OHPT hypothesis in the type-A homonym list, in order to improve our theorems of semantic relatedness.

Using the Italian translations of MultiWordNet, we discovered 16 homonymous words (out of 1603) violating OHPT, including two cases due to MultiWordNet errors, four cases related to homonym mapping errors that we manually fixed afterward, and 10 cases of parallel homonyms between English and Italian. Most of the latter 10 cases were cognates, based on their orthographic similarities. These parallel homonyms and their two definitions are in Table 5.3. In each row of this table, both the English and the Italian words cover both of the homonymous meanings and therefore are parallel homonyms.

In the second step, we extend our experiments to use the translation information from BabelNet. Having more than 280 languages, BabelNet provides us with 154 cases of OHPT violations. An analysis of these cases required dictionary information in many languages, which was not easily available. However, we could observe that languages closer to English create the majority of these semantic bridges between homonymous senses. For instance, Irish and French together account for more than 30% of these 154 cases. This percentage suggests that a great number of these cases are apparent parallel

homonyms.

A more formal evaluation of the OHPT hypothesis and its corollary theorem is our next step. To be able to evaluate the strength of the OHPT method, we try to classify words of our balanced development set into homonymous and polysemous, using Theorem 1. Starting with Italian translations from BabelNet, we achieve an F_1 score of 69% with 54% precision and 95% recall. The low precision and the high recall suggests that most of the words are classified as homonymous. Having so many words classified as homonymous means that very limited evidence for semantic relatedness can be obtained from only the Italian translations of BabelNet. Therefore we decided to extend our experiments with more languages.

5.2.2 Optimizing on Languages

In the next step, we expand the experiments by incorporating the translation information of different languages, since different languages have different structures of meaning, which can have different levels of similarity to that of the English language. On the other hand, the coverage of different languages in BabelNet can be an effective factor in the results of our experiments.

As the first step, a list of the top 50 languages in BabelNet is made, ordered by the number of synsets. Then the performance of each of these languages is evaluated by providing the translations of that language to our OHPT method for homonym detection. The resulting F_1 scores of these experiments provide a ranking of the languages, which can roughly be interpreted as the informativeness of BabelNet translations of those languages for our task. Figure 5.1 illustrates the top 10 languages by their F_1 score in detecting homonymous words, using OHPT. We do not include the rest of the languages in the figure, as the trends in the changes of precision, recall, and F_1 score are similar to the first 10 languages, meaning that the precision and the F_1 score generally decrease, and the recall increases. The increase in recall means more words are detected as homonyms, which in turn means fewer semantic relations can be detected as a result of insufficient evidence. It is important to note that an increase in recall would be desirable when it is resulted by the

absence of *incorrect* translations. However, in the above comparison of the languages, the simultaneous decrease of the precision suggests a general deficiency of translations, which is not desirable. Interestingly, similar languages such as Indonesian and Malay were similar in their ranking. This can be seen for Spanish, Catalan, and Portuguese, as well. Moreover, we expect one of the main reasons for Indonesian ranking first to be its distance from English. That is to say that we expect distant languages to perform better as they share fewer parallel homonyms with English. On the other hand, the coverage of the senses in BabelNet is another important factor in the ranking of the languages, because of the negative effect of the missing translations.

Our second step is to combine the translations of different languages when providing them to our model. Since parallel homonymy is an exception to our theorems, we prefer to combine dissimilar languages, which are less likely to contain parallel homonyms. Therefore, we first exclude Malay as it is a variant of Indonesian, and then, Catalan, Portuguese, Romanian, Croatian, and Italian are to be removed, since they are all European languages similar to Spanish. As a result, we work with the five languages of Indonesian, Spanish, Slovenian, Finnish, and Japanese. We try an iterative approach of adding these languages to the combination, according to their ranking, which is based on their independent performance. As demonstrated in Figure 5.2, the increase in the number of languages has a positive effect on the precision of the detected homonyms. At the same time, the number of true homonyms that are discovered decreases, which based on our previous analyses can be because of the noisy translations or the parallel homonyms that are introduced by the new languages. Finally, the best F_1 score is 76.4% for the second step, which is the combination of Indonesian and Spanish.

5.2.3 Adding Other Semantic Relation Methods

We continue our experiments with our other semantic relation theorems, trying them individually and combined together on the development data. Table 5.4 presents the development results of using our semantic relatedness theorems with the Indonesian translations from BabelNet and the combination of both

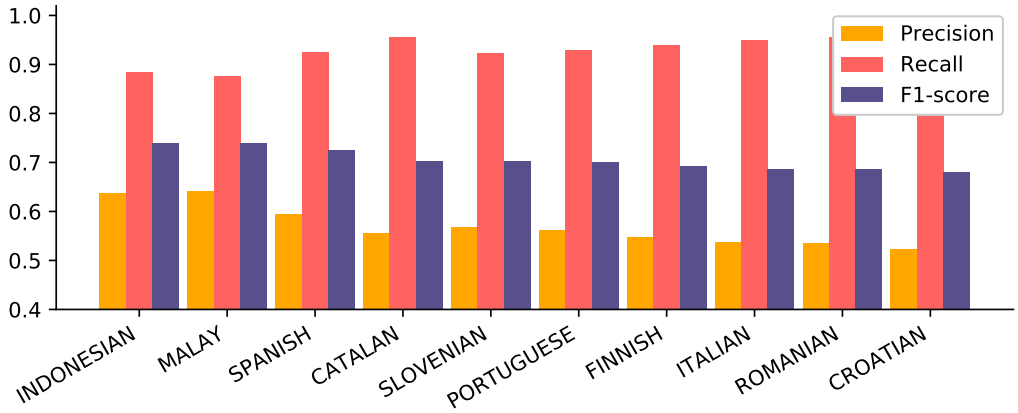

Figure 5.1: The 10 best languages in terms of F_1 score in homonym detection, out of 50 languages with the most number of synsets in BabelNet.

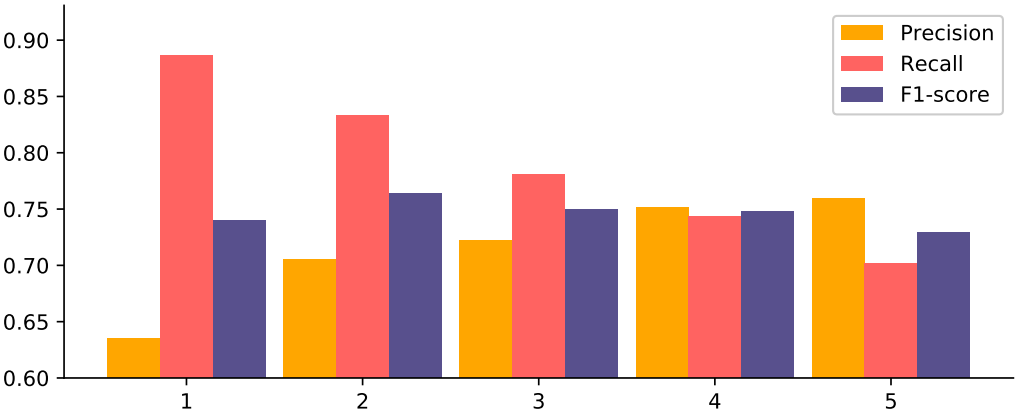

Figure 5.2: A comparison of the use of different number of languages in using OHPT for homonym detection on the balanced development set. At each step, a language is added from Indonesian, Spanish, Slovenian, Finnish, and Japanese.

Indonesian and Spanish translations. Here is a description of each method:

- The first method is a simple baseline. The idea of this baseline is that homonymous words are expected to have a higher number of senses in any sense inventories. Therefore, the baseline method is to consider any words with more than k senses in WordNet to be homonymous, and otherwise polysemous. The value of k is tuned and the best F_1 score belongs to $k = 1$, which means that any word with more than one sense is classified as homonymous, and in our balanced data, that would be all words. Hence the 50% precision and 100% recall. It is important to note that a more informed baseline can use the path similarity in WordNet's hypernymy graph; however that is part of the method of van den Beukel and Aroyo (2018), which will be compared to our test results in the followings.
- The second method is OHPT, which was discussed in the previous section. This method provides a considerable improvement over the baseline in both data settings.
- The third method (2T) belongs to Theorem 2 (2Words-2Translations). The results of this method demonstrate improvements over the OHPT method in terms of precision.
- The fourth method (2S) belongs to Theorem 3 (2Synsets-2words). Similar to 2T, there is considerable improvement of results over the baseline; however, the improvement over OHPT is slight.
- The fifth method (3M) belongs to Theorem 4 (3Multisynsets), and it displays similar results to 2S in terms of precision, with slightly better recall, which leads to better F_1 score.
- The next method (HM) is that of section 4.2.7 (Hypernymy&Hyponymy), which is not as strong as the other methods. The precision value for this method is 50.3%, which is slightly higher than the baseline. That means not many senses of the same word are sister terms in hypernymy and meronymy, yet if they are, they would belong to the same homonym.
- The seventh item belongs to the combination of the Theorems 2 and 3, which are complementary to each other, as discussed in Chapter 4.

data	Indonesian			Indonesian+Spanish		
	Pre	Rec	F_1	Pre	Rec	F_1
Baseline	50.0	**100**	66.7	50.0	**100**	66.7
OHPT	63.5	88.7	74.0	70.5	83.4	76.4
2T	66.4	86.0	74.9	74.3	78.4	76.3
2S	64.3	87.9	74.3	71.0	82.9	76.5
3M	64.3	91.0	75.3	70.7	83.9	**76.7**
HM	50.3	**100**	67.0	50.3	**100**	67.0
2T+2S	67.7	85.8	75.7	74.8	77.6	76.2
2T+2S+3M+HM	**68.6**	84.7	**75.8**	**74.9**	75.7	75.3

Table 5.4: Development results for the comparison between the use of different semantic relation methods on the Indonesian translations from BabelNet and the translations in two languages of Indonesian and Spanish from BabelNet.

This combination provides improvements in precision over the previous methods in both settings and increases the F_1 score for the Indonesian data.

- The combination of all four new methods is the last item, which produces the best precision for both settings because it retrieves the most semantic relations, which would eliminate more false-positive cases. However, the F_1 score is the highest only for the first data setting, which can be explained by the lower recall, indicating a higher number of false-negatives.

In conclusion, we choose the combination of the four methods of 2Translations, 2Synsets-2Words, 3Multisynsets, and Hypernymy&Hyponymy as our best method in the development process, as it results in the best precision for both language settings and the best F_1 score in the use of one language. Our preference towards a higher precision, rather than recall, is because detecting *correct* homonyms is a priority for us, and we want to discover any possible evidence for polysemy.

Finally, we test our best method and the OHPT method on our balanced test set. As demonstrated in Table 5.2.3, both the OHPT and the combined method perform superior to the baseline and the previous VDB method, which uses sense definitions (van den Beukel and Aroyo, 2018). We achieve the best F_1 score of 78.6% for the OHPT and the best precision of 79.1% for the combined method. The accuracy follows the same trends as the F_1 score.

The higher recall of the baseline and the VDB method implies their maximal effect on considering most of the words to be homonymous and most senses unrelated, which can also be observed in the number of true-positives and false-positives. In this table, the homonymous words are considered to be positive, and the polysemous words negative.

Furthermore, we test our best method on our new homonym resource, which we have created using the list of homonyms of Rice et al. (2019). We tried our OHPT method on this list using Persian translations from BabelNet. Out of the 94 homonymous words, we have found evidence for the polysemy of 20 words. In other words, we have found Persian translations shared between different lexemes of these 20 English words. However, after manual evaluation of these translations, we have found all of them to be incorrect and due to noise in BabelNet. We then tried French translations, which yielded to one piece of correct evidence against the homonymy of the word *drone*. The two senses of *bee* and *monotone voice* of this word, both share the French translation *bourdon*, which indicates a semantic relation between these two senses. Therefore, we claim that the two above senses of *drone* are not homonyms of each other.

	Pre	Rec	F_1	Acc	TP	TN	FP	FN
Baseline	50.0	**100**	66.7	50.0	95	0	95	0
VDB	51.1	99.0	67.4	52.1	94	5	90	1
OHPT	74.5	83.2	**78.6**	**77.4**	79	68	27	16
2T+2S+3M+HM	**79.1**	71.6	75.1	76.3	68	77	18	27

Table 5.5: Test results of homonym detection for OHPT and the combined method (2T+2S+3M+HM) along with the baseline and van den Beukel and Aroyo's method (VDB). The result columns, from left to right, represent precision, recall, F_1 score, accuracy, true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN).

5.2.4 Results on Semantic Relation Detection

Our previous evaluations were on a word-level task of homonym detection, which was the end-task. However, the immediate result of our models is on determining semantic relatedness between pairs of senses, and then the de-

cision on the entire word follows. The immediate results are much easier to interpret. Because the positive cases are related senses, and the negative cases are unrelated senses. This interpretation is more intuitive and is the opposite of what we had in the end-task experiments, where the positive meant homonymous (i.e. having unrelated senses) and the negative meant polysemous (i.e. having only related senses).

The test results of semantic relation detection between pairs of senses are in Table 5.6. The combined method (2T+2S+3M+HM), does best in the recall (80.5%), which here means the ability of the model to retrieve as many sense relations as possible. On the other hand, the precision (81.8%) is the lowest for this model, and that means the proportion of the incorrect relations was higher than the other two models. However, this difference in precision is not as significant as that of the recall, which shows its effect in the F_1 score, where the best F_1 score (81.1%) is still for the combined method. It is important to note that we would prefer recall, here, over precision, because we want to retrieve as many semantic relations as possible. Another point to consider is that the performance difference between our methods and the VDB method is even more visible in the current evaluation compared with the end-task evaluation of homonym detection in Table 5.2.3. Moreover, Table 5.6 also includes the exact number of correct and incorrect predictions of each model within the 1514 sense pairs, which include 1060 related and 454 unrelated sense pairs. Here, our combined method achieves the highest number of semantic relations retrieved (TP) and the lowest number of semantic relations missed (FN).

The OHPT model is the best in the word level, where the combined model performs best in the sense level. We consider the latter a more accurate measure of the strength of a model, even though our end task is on the word level because different words have different numbers of sense pairs and even a mistake in one of the sense pairs would classify the entire word incorrectly. Therefore, we take the combined model as our best model in conclusion.

	Pre	Rec	F_1	Acc	TP	TN	FP	FN
VDB	**92.2**	22.3	35.9	44.3	236	434	20	824
OHPT	86.5	65.8	74.7	68.8	697	345	109	363
2T+2S+3M+HM	81.8	**80.5**	**81.1**	**73.8**	853	264	190	207

Table 5.6: Test results on semantic relation detection for OHPT and the combined method (2T+2S+3M+HM) compared with van den Beukel and Aroyo's method (VDB). The result columns, from left to right, represent precision, recall, F_1 score, accuracy, true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN).

5.2.5 Experiments with Random Walks

In our previous experiments, we assumed the transitivity of the semantic relatedness only within the concepts of a word, as discussed in Chapter 4. In other words, we created separate graphs for each word, and two concepts would have been related if there was a path between them in such a graph. However, this transitivity assumption can be extended across different words by considering a unified graph for all of the concepts in the sense inventory. In such a graph, if senses of a word appear in the same connected component, they are considered semantically related, even if they did not have a direct connection through any of our semantic relatedness methods. In this case, if our methods fail to find a direct semantic relation between two concepts, an indirect path of relations through the concepts involving other words will compensate for that failure. Therefore, this graph should ideally have senses of the same lexeme in the same component, and on the other hand, senses from different lexemes of a word should be in separate components.

Our initial experiments with this idea faced two main difficulties. One particular problem was the extremely unbalanced size of the connected components. A large number of concepts formed a single large connected component, with the rest of the concepts having very small connected components. The large connected component was formed due to errors propagating through the graph. In other words, every single incorrect translation that formed a semantic relation between two irrelevant senses, could merge a significant number of irrelevant senses into a connected component, and when that was repeated for

a number of times with other incorrect translations, the connected component of irrelevant senses would have grown even more. On the other hand, the small connected components were due to the sparsity of translations in the synsets of those components.

The problem of small connected components did not have any simple remedy, yet, the large connected components could have been handled by algorithms that are resistant to noise. In particular, we hoped to cancel the effect of noise in connecting senses through random walks. In this approach, to check the relatedness of two synsets, we ran several iterations of random walks, each time for a certain number of steps, and considered the average number of times that the target synset was visited as a measure of its relatedness to the starting synset. However, this process was highly time-consuming and depending on the number of the edges of the graph and the parameters of the random walk, it could take as long as several hours to be completed for a single word. Eventually, we could not achieve significant results from this approach in our limited time frame. Nonetheless, we are optimistic that further experiments and exploration of this approach can produce adequate results in retrieving semantic relations and, eventually, the task of homonym detection.

5.3 Discussion on Errors

We demonstrate promising results on the evaluation of our methods of homonym detection on a set of balanced gold-standard data. In this section, we try to provide an analysis of the error cases. Our best method for homonym detection (i.e. the OHPT method) on the test set of 190 words makes 43 errors, including 27 false-positives and 16 false-negatives.

5.3.1 False-Positives

The false-positive cases (i.e. words incorrectly detected as homonymous) are the cases that we could not discover any evidence for semantic relatedness between the senses of a word, primarily due to the sparsity of BabelNet synsets. When working with one language, as we have started with only Italian trans-

lations, there will be many senses of English words that do not have any translations in that one language. We tried to remedy this by using multiple languages, which would not entirely eliminate this problem. An absence of translations is either because of lexical gaps or the resource simply not being complete. Lexical gaps are where a word cannot be translated into a single word in the target language. As an example of lexical gaps, we can mention the English word $seamless_a$ with these three senses:

1. $seamless_a^1$: Not having or joined by a seam or seams.
2. $seamless_a^2$: Smooth, especially of skin.
3. $seamless_a^3$: Perfectly consistent and coherent.

The first sense is a "lexical gap" in Italian, according to MultiWordNet. Thus, if we are to consider Italian translations, we would never be able to infer the semantic relatedness of this sense to the other two senses.

Another point to consider is that in the process of building multi-wordnets, such as BabelNet, the structure is that of an English WordNet, with a certain set of concepts, which is not necessarily similar to other languages. Non-English wordnets such as MultiWordNet are usually built by adding translated senses to synsets from Princeton WordNet when a translation is available and then adding extra synsets when necessary(Hauer and Kondrak, 2020b). This would create a bias towards the English structure of meaning. One of the results is that there are concepts from some languages that are not present in BabelNet simply because there were no English words that would translate to them. These Persian words, for example, are not present in BabelNet:

- پریروز/pæriːˈruːz/ : The day before yesterday.
- قهقهه/ɢæhɢæˈhe/ : A loud laugh.
- تعارف/tæɒːˈrof/ : A particular Iranian art of etiquette.

5.3.2 False-Negatives

The false-negative cases are the homonyms that we have discovered evidence suggesting semantic relatedness between their unrelated senses. These happen for three reasons:

Noise in translation resource

The incorrect translations in BabelNet were a source of many errors in our semantic relations. BabelNet, in particular, is built through the integration of different sense inventories and some of these are not manually created. As an example of such errors, the word $shark_v$ can be mentioned that has a BabelNet synset bn:00093576v meaning "to trick someone", but it has the incorrect French translations of "requin" which means "the animal shark". Another example is bn:00008576n for the word "lighter" that means "A flatbottom boat for carrying heavy loads" but is translated to a Persian word کبریت/keb'ri:t/ that is related to the other sense of "lighter" meaning "match".

Another form of noise is the inexact translations that are added to the multi-synsets. For example, the synset bn:00084061v with the word $bring_v$ defined as "Go or come after and bring or take back" is translated to the Persian verb بردن/bor'dæn/ meaning "To take something/someone to another place", which does not have the meaning of "coming back" but only "going".

Noise in the gold data

Our gold sense mapping of the type-A homonyms is using the ODE clustering from Navigli (2006) that is automatically built and has some senses incorrectly clustered together. Therefore, if an incorrect cluster of senses is mapped to a homonym based on one of its members only, then the other members can have an incorrect mapping. For example, the word "content" has two lexemes with the meanings of "satisfaction" and "load". We found these two senses of $content_n$ clustered together:

1. $content_n^5$: The sum or range of what has been perceived, discovered, or learned.
2. $content_n^6$: The state of being contented with your situation in life

The distinction between these two senses is clear to a human judge; however, this error leads to both senses being mapped to the same homonym meaning "satisfaction". As mentioned earlier, we corrected these cases upon finding them, yet, since we could not revise the entire homonym mapping resource,

we assume there are more of these error cases present in the data.

Parallel Homonyms

As discussed in Section 2.6 parallel homonyms are exceptions to our methods of semantic relatedness. To be able to analyze the role of parallel homonymy in our errors, we ran our OHPT method on the test set using translations from Persian. The model could achieve a precision of 51%, recall of 90%, and F_1 score of 64%. We believe that even though these results are considerably worse than the results of the best language setting, an analysis of the proportion of errors in this language can still be informative.

The OHPT model using Persian translations had 49 false-negatives and 416 false-positives out of the 948 words in the test set. Our analysis suggests only two of these error cases to be due to parallel homonymy. Here are the two concepts creating the first error:

- *bn*:00011824*n* : The temperature at which a liquid boils at sea level.
- *bn*:00011823*n* : A painful sore with a hard core filled with pus.

Both of these concepts are represented in English by the word "boil" and in Persian by the word جوش/d͡ʒuːʃ/. The second error happens between these two concepts:

- *bn*:00021722*n* : An association of sports teams that organizes matches.
- *bn*:00050410*n* : An obsolete unit of distance of variable length.

Both of these concepts are lexicalized by the English word *league* and the Persian word لیگ with the same pronunciation. All of the other 47 cases of false-negatives are due to incorrect translations from BabelNet, which means only a 4% of the false-negatives and less than 1% of the entire error cases are due to parallel homonymy. This is evidence of the rareness of this phenomenon. Moreover, excluding the incorrect translations improves the results. The precision becomes 53% with the recall of nearly 100%, and the F_1 score of 69%.

The role of parallel homonymy in the errors can be different depending on the similarity of the target languages to the English language. Nevertheless, based on the above analysis, we can assume that the majority of the errors

have causes such as synset sparsity and noisy translations.

Chapter 6

Conclusion

In this thesis, we introduced novel methods of semantic relation detection and performed experiments on distinguishing between homonymy and polysemy, utilizing translation information. Our approach was to build graphs of semantic relations between senses of words and considering the senses in the same connected component as polysemous. Through this approach, we presented state-of-the-art results on the task of homonym detection.

We furthered our experiments by building a large graph that considers semantic relatedness across different words. However, due to noisy translations, this extension led to a problem, which was the construction of an excessively large connected component containing unrelated concepts. A future direction to solve this problem is to use random walks on this graph to achieve a similarity measure between the concepts. This approach can be beneficial because random walk algorithms are resistant to noise. Another direction is to use graph embeddings to compare the similarity of concepts. Both of these approaches can eventually help to distinguish between homonymy and polysemy.

The other contributions of this work are two homonym resources, one created and another updated. These resources have been utilized in the current research and can be beneficial for future research on homonymy, sense inventories, and semantic relatedness.

References

Marianna Apidianaki. 2008. Translation-oriented word sense induction based on parallel corpora. In *Language Resources and Evaluation (LREC)*.

Mohit Bansal, John DeNero, and Dekang Lin. 2012. Unsupervised translation sense clustering. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782.

Barend Beekhuizen, Sasa Milic, Blair C Armstrong, and Suzanne Stevenson. 2018. What company do semantically ambiguous words keep? insights from distributional word vectors. In *CogSci*.

Luisa Bentivogli and Emanuele Pianta. 2000. Looking for lexical gaps. In *Proceedings of the ninth EURALEX International Congress*, pages 8–12. Stuttgart: Universität Stuttgart.

Sven van den Beukel and Lora Aroyo. 2018. Homonym detection for humor recognition in short text. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 286–291.

Jordi Daudé, Lluis Padro, and German Rigau. 2000. Mapping wordnets using structural information. *arXiv preprint cs/0007035*.

Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. 2017. Eurosense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 594–600.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Helge Dyvik. 2004. Translations as semantic mirrors: from parallel corpus to wordnet. In *Advances in corpus linguistics*, pages 309–326. Brill Rodopi.

Helge Dyvik. 2009. Semantic mirrors. *Unpublished manuscript*.

Christiane Fellbaum. 1998. WordNet: An on-line lexical database and some of its applications. *MIT Press*.

Kyle Gorman, Gleb Mazovetskiy, and Vitaly Nikolaev. 2018. Improving homograph disambiguation with supervised machine learning. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Arnob Mallik, and Grzegorz Kondrak. 2020a. Low-resource G2P and P2G conversion with synthetic training data. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–122, Online. Association for Computational Linguistics.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Arnob Mallik, and Grzegorz Kondrak. 2020b. Ualberta at semeval-2020 task 2: Using translations to predict cross-lingual entailment. In *Proceedings of the SemEval-2020 International Workshop on Semantic Evaluation*.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Rashed Rubby Riyadh, and Grzegorz Kondrak. 2019. Cognate projection for low-resource inflection generation. In *Proceedings of the 16th Workshop on Computational*

Research in Phonetics, Phonology, and Morphology, pages 6–11, Florence, Italy. Association for Computational Linguistics.

Bradley Hauer and Grzegorz Kondrak. 2020a. One homonym per translation. *AAAI-2020*.

Bradley Hauer and Grzegorz Kondrak. 2020b. Synonymy= translational equivalence. *arXiv preprint arXiv:2004.13886*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

Daniel Jurafsky and James H Martin. 2008. *Speech and Language Processing*. Prentice Hall.

Grzegorz Kondrak et al. 2020. Detection of polysemy in multi-wordnets. *In preparation*.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Daniel Mirman, Ted J Strauss, James A Dixon, and James S Magnuson. 2010. Effect of representational distance between meanings on recognition of ambiguous spoken words. *Cognitive Science*, 34(1):161–173.

Roberto Navigli. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 105–112.

Roberto Navigli. 2018. Natural language understanding: Instructions for (present and future) use. In *IJCAI*, pages 5697–5702.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *First international conference on global WordNet*, pages 293–302.

Lonneke Van der Plas and Jörg Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 866–873. Association for Computational Linguistics.

Caitlin A Rice, Barend Beekhuizen, Vladimir Dubrovsky, Suzanne Stevenson, and Blair C Armstrong. 2019. A comparison of homonym meaning frequency estimates derived from movie and television subtitles, free association, and explicit ratings. *Behavior research methods*, 51(3):1399–1425.