A FLEXIBLE COMPOSITIONAL APPROACH TO WORD SENSE DISAMBIGUATION

ALEX DE PAULA BARROS

A FLEXIBLE COMPOSITIONAL APPROACH TO WORD SENSE DISAMBIGUATION

Dissertação apresentada ao Programa de Pós-Graduação em Computer Science do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Computer Science.

Orientador: Nivio Ziviani
Coorientador: Adriano Alonso Veloso

Belo Horizonte
Julho de 2018

ALEX DE PAULA BARROS

A FLEXIBLE COMPOSITIONAL APPROACH TO WORD SENSE DISAMBIGUATION

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Nivio Ziviani
Co-Advisor: Adriano Alonso Veloso

Belo Horizonte
July 2018

Barros, Alex de Paula

B277f A Flexible Compositional Approach to Word Sense Disambiguation / Alex de Paula Barros. — Belo Horizonte, 2018
xx, 41 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais — Departamento de Ciência da Computação.
Orientador: Nivio Ziviani
Coorientador: Adriano Alonso Veloso

1. Computação – Teses. 2. Recuperação da informação. 3. Processamento de linguagem natural (Computação). I. Orientador. II. Coorientador. III. Título.

CDU 519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

A Flexible Compositional Approach to Word Sense Disambiguation

ALEX DE PAULA BARROS

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. NIVIO ZIVIANI - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ADRIANO ALONSO VELOSO - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. FLAVIO VINICIUS DINIZ DE FIGUEIREDO
Departamento Ciência da Computação - UFMG

PROF. RENATO ANTÔNIO CELSO FERREIRA
Departamento de Ciência da Computação - UFMG

PROF. WLADMIR CARDOSO BRANDÃO
Departamento de Ciência da Computação - PUC/MG

Belo Horizonte, 27 de julho de 2018.

Acknowledgments

I'd like to thank those who gave me strength to move forward, my wife Fernanda, all of my friends at Kunumi and the company itself. I'd like to thank my advisors for the opportunities they gave me and for showing me the way, and most of all, thank the Lord for His infinite grace.

"For from him and through him and to him are all things. To him be glory forever. Amen."

(Romans 11:36)

Abstract

Word sense disambiguation is identifying which sense of a word is used in a sentence when the word has multiple meanings. The most successful algorithms to date use supervised machine learning. In these methods, a classifier is trained for each distinct word on a corpus of manually sense-annotated examples. One possible drawback is their lack of flexibility due to requiring annotated examples for potentially every word in the vocabulary. In contrast, knowledge-based methods do not require a classifier for each distinct word and are often built over lexico-semantic resources like ontologies, thesauri or machine-readable dictionaries. In this work, we propose a flexible compositional algorithm based on context-gloss comparisons, that compares local context of a word represented by its neighbor words with glosses of the possible senses a word can assume using a semantic distance measure. The algorithm has three components, each based on a different information source: (i) sense frequency, obtained by counting the number of times a word occurs with each meaning in an annotated corpus, (ii) extended gloss, obtained by expanding a word dictionary definition using related words in an ontology (e.g., car and automobile), and (iii) sense usage examples, obtained from inventories that provide sentences with usage examples for some senses. Our compositional approach is flexible in the sense that it is not solely dependent on annotated examples and works well even when some or all of the three aforementioned knowledge sources are not available. We evaluated the performance of our algorithm for all possible combinations of the three components, simulating different scenarios of knowledge sources availability. The algorithm achieves an F1 score of 67.5 when all components are available, presenting a favorable result when compared with a state-of-the-art knowledge-based system that achieves an F1 score of 66.4.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

Natural language is inherently ambiguous. From a technical standpoint, any sentence in a language with a large-enough grammar can have alternate plausible interpretations and denotations, as most commonly-used words have several senses or meanings. As a result, the particular meaning of a word on a particular occasion is often determined by its context. The word 'light', for instance, have many possible meanings, including 'illumination' and 'not heavy'. Still, in the sentence 'She turned on the light', it is clear the sense in play. In the following sentences: 'They are going on a school trip' and 'To trip over a tree branch', the word 'trip' assumes two different senses, 'travel' or 'excursion' in the first sentence and 'misstep' in the second. Ambiguity does not have a well-defined solution, and thus while humans usually do not need to think about the multiples senses a word has to identify the underlying meaning, a machine needs to process additional sources of unstructured textual information to achieve the same goal. According to [Navigli, 2009]), the identification of meanings of words in context is called word sense disambiguation (WSD, hereafter).

WSD has been a core research topic in Natural Language Processing (NLP) for many decades and has received considerable interest over recent years. This interest is justified since effectively dealing with lexical ambiguities can benefit a variety of downstream tasks, such as Information Retrieval and Information Extraction (Zhong and Ng [2012]; Bovi et al. [2015]), Machine Translation (Carpuat and Wu [2007]; Xiong and Zhang [2014]; Neale et al. [2016]), Question Answering (Peters et al. [2018]). WSD has been addressed with a variety of methods, from graph-based solution (Moro et al. [2014]) to learn continuous vector representations for word senses (Chen et al. [2014]; Iacobacci et al. [2015]; Flekova and Gurevych [2016]).

Existing WSD solutions are grouped into two main categories: supervised (or semi-supervised) and knowledge-based (unsupervised). Supervised solutions employs sense-annotated corpora created by expert annotators in order to train models that can disambiguate words. The effectiveness of supervised models increases with the amount of sense-annotated data available, and this is a limiting factor as the labeling process may render a fully labeled training set infeasible. Unlike supervised models, knowledge-based models require only external knowledge sources, which in turn, are not always available or when available are not complete enough to disambiguate some words.

In practice, supervised models consistently outperform knowledge-based approaches (Raganato et al. [2017b]), but these models have limitations which are related to sense-annotated corpora required for training, as follows:

- The first limitation is the difficult to create sense-annotated corpora when non-expert annotators are involved (de Lacalle and Agirre [2015]), which makes the training corpora more expensive to create.

- The second is related to words covered by the sense-annotated corpora since supervised approaches are only able to disambiguate words that were annotated in the train corpora, it limits supervised models performance in all-word WSD tasks.

- The third is related to the need of a dedicated classifier (word expert model), which needs to be trained for every target lemma, making them less flexible than knowledge-based approaches and hampering their use within end-to-end applications. A recent study addresses this issue using sequence learning models (Raganato et al. [2017a]), removing the necessity of a dedicated classifier, therefore providing a more flexible supervised system with a small decline in accuracy, but it was still bounded by shortage of sense-annotated corpus.

Unlike supervised models, knowledge-based systems usually require external knowledge sources such as WordNet (Miller [1995]). These systems are built upon the structural properties of lexico-semantic resources (Agirre et al. [2014]; Moro et al. [2014]; Weissenborn et al. [2015]), and some of them have the ability to handle multiple target words at the same time and disambiguate them jointly (Lesk [1986]; Banerjee and Pedersen [2002]). They also have the capacity to disambiguate all words in its underlying lexicon, thus being not limited by the sense-annotated examples present in a training corpus.

One possible drawback of knowledge-based approaches is the limited availability of knowledge sources, as well as their incompleteness. For instance, WordNet is a resource used by many solutions as their primary, if not their only, knowledge source. However, although similar resources are available in languages other than English, it may not contain all kinds of information present in the English version. For example, a Brazilian version of WordNet (Dias-da-Silva et al. [2008]) contains information about word relations and sense usage examples but does not provide sense frequency, information which is so important that it is explored by all baseline algorithms we considered to compare with our proposed solution.

1.2 Our Solution

We designed and implemented a flexible algorithm for the WSD problem. For that, we provide a solution for scenarios where some sources of information may be unavailable. Our solution uses the idea proposed by Lesk [1986], which introduced the gloss-context comparison and inspired various extensions of the original algorithm (Banerjee and Pedersen [2002]; Basile et al. [2014]). Lesk [1986] made one of the first attempts to use a machine readable dictionary for WSD. Lesk used definitions in the dictionary to express the way in which the choice of one sense in a sentence was dependent upon the senses chosen for words close to it. He used the Oxford Advanced Learner's Dictionary [Hornby and Crowther, 1963] by choosing the senses which share definition words with senses from neighbouring words in a sentence.

Our proposed algorithm is thus called Compositional Lesk Algorithm (CLA). It is compositional because it employs a structured combination of different components, each component providing distinct but complementary sources of information. More specifically, our proposed CLA algorithm selects the most appropriate sense for the context where the word appears based on the comparison between the context of the word (e.g., gloss of each sense candidate) and the possible meanings which the word can assume (e.g., list of meanings in a dictionary).

Components: The components of the proposed solution are based on the availability of three sources of information: (i) sense frequency, obtained by counting the number of times a word occurs with each meaning in an annotated corpus, (ii) extended gloss, obtained by expanding a word dictionary definition using related words in an ontology (e.g., car and automobile), and (iii) sense usage examples, obtained from inventories that provide sentences with usage examples for some senses. These information sources are commonly used by knowledge-based solutions, but when unavailable they hinder

and sometimes make impossible the use of some solutions. For instance, Raganato et al. [2017b] shows that state-of-art knowledge-based system uses sense frequency as the only criteria of selection of the most appropriate sense in 63% of cases. On contrary, our solution builds each component based on the aforementioned information sources and perform WSD effectively even when some or all components are missing. In other words, our proposed CLA algorithm is flexible in the sense that it is not dependent on annotated examples and is likely to work well even when some (or even all) of the three aforementioned knowledge sources are unavailable.

From the three components of the proposed algorithm, two of them, extended gloss and sense usage examples, are used to improve the quality of sense candidate's gloss, while sense frequency is used to calculate the probability of a word assuming each sense and this probability is used to weight the semantic distance. We evaluated every combination of components, simulating different scenarios of information availability and show the contribution of each component to WSD performance in different scenarios.

Semantic Space: We assume that semantically similar words tend to have similar contextual distributions, and thus word embeddings are likely to benefit WSD. Word embeddings represent words in a low-dimensional continuous vector space. These vectors capture useful syntactic and semantic information from contextual regularities in language. Word embeddings serve as input features and help overcoming the word sparseness of natural texts, being thus effective to compute the similarity between the gloss of a sense and the context of the word.

Our proposed CLA algorithm explores Word Mover Distance (WMD, Kusner et al. [2015]) as the semantic distance measure to make context-gloss comparisons. WMD is based on the distance a sentence or document must "travel" in the semantic space to be equal to other sentence or document. As this semantic distance measure built on word embeddings, we also evaluated the impact of each information scenario using two different pre-trained word embeddings models: Global Vectors (GloVe) [Pennington et al., 2014], which is based on matrix factorization and Skip-gram [Mikolov et al., 2013], which is based on artificial neural networks.

1.3 Contributions

In summary, our main contributions are as follows:

- We propose a flexible compositional approach for WSD that, unlike the baselines

presented in Section 4.2, is able to perform word sense disambiguation in different scenarios of knowledge sources scarcity. We show that when all components are available CLA achieves an F1 score fo 67.5, which is comparable with a state-of-art knowledge-based solution whose result is an F1 score of 66.4.

- We evaluated each scenario of information availability, showing the importance of each component, how they interact with each other and the impact of each component in the algorithm performance. We show that comparing the scenario where none of the components are available to a scenario where there is only a component is available, sense frequency improves the F1 score by 19.4, while extended gloss provides an improvement of 9.2 and sense usage examples 1.8.

- As the semantic distance used in CLA is built on distributed word representations, we evaluated each scenario of knowledge source availability using two embedding models, GloVe [Pennington et al., 2014] and Skip-gram [Mikolov et al., 2013]. We show that both embeddings achieved similar performance when all components are available with a slight difference in different scenarios.

1.4 Organization

The rest of this dissertation is structured as follows. In Chapter 2, we present the description of the WSD task and discuss related work. In Chapter 3, we describe our proposed algorithm and its components. In Chapter 4, we describe the datasets and the baselines used in the experiments and the experimental results. Finallly, in Chapter 5, we present concluding remarks and future work.

Chapter 2

Background and Related Work

2.1 Task Description

Natural language processing (NLP) is a broad field of science which comprises a number of tasks, such as part-of-speech tagging, named entity recognition, text classification, text summarization. Word sense disambiguation is also an NLP task, it can be seen as a classification task, but with the difference that each word has a set of classes (senses) while other NLP classification tasks have a single predefined set of classes. WSD can be viewed as n distinct classification tasks, where n is the vocabulary size.

The generic WSD task has two variants:

- Lexical sample (or target WSD): disambiguate a set of target words, usually with only one occurrence per sentence.

- All-words WSD: disambiguate all open class words in the text (i.e., nouns, verbs adjectives, and adverbs).

Word sense disambiguation can formally be described as the task of identifying the appropriate sense(s) of all words in a text T, where T is a sequence of words $T = (w_1, w_2, w_3, ..., w_n)$ (disregarding punctuation) and a *sense* or *wordsense* is one of the meanings of a word. In other words, WSD can be defined as the task of identifying a mapping A from a set of words to senses, such that $A(i) \subseteq Senses_D(w_i)$, where $Senses_D(w_i)$ is the set of senses in a dictionary D for word the w_i, and $A(i)$ is the subset of the senses of w_i which are appropriate in the context T. Although mapping A can assign more than one sense to each word $w_i \in T$, typically only the most appropriate sense is selected, that is, $|A(i)| = 1$.

2.2 WSD Elements

WSD task usually has four elements: sense selection, external knowledge sources, context representation, and classification method.

2.2.1 Sense Selection

The sense selection in WSD refers to the selection of the senses from which the WSD system will disambiguate, in other words, the selection of classes for a target word. Consider the following example:

- Fernanda chopped the vegetables with her **knife**.
- A guard was stabbed with a **knife** by the invader.

In this example, the two senses refer to the same object but differ in their usage, first as a kitchen tool and latter as a weapon. The word senses provided will depend on the sense inventory (i.e., dictionary, encyclopedia or other resource that provides a list of senses for a word type) chosen. For instance, a sense inventory may provide a single sense for the word knife ('a blade for cutting') while others may distinguish between tool and weapon, and therefore provide two senses for the word knife. The sense inventory selection should be defined based on the intended application. Moreover, words senses cannot be easily reduced to a finite set of entries because languages are everchanging and subject to interpretation (e.g., 'meat' once meant solid food in general).

Granularity is another aspect of sense selection. In WSD granularity refers to how specific is the distinction of word senses. In WordNet (Miller [1995]), for example, the word *snow* most frequent senses are:

- precipitation falling from clouds in the form of ice crystals
- a layer of snowflakes (white crystals of frozen water) covering the ground'

While falling from the sky snow holds the first meaning, it changes to the second sense when it reaches the ground. The task that aims to discriminate between these senses is called fine-grained while the task that doesn't is called coarse-grained.

2.2.2 External Knowledge

Knowledge sources are fundamental in WSD. They are essential to associate senses with words (e.g., dictionaries are used to create gold tagged datasets). There are structured

(e.g., thesauri, ontologies, and machine readable dictionaries) and unstructured (e.g., raw text corpora) knowledge sources. Resources like lists of *stopwords* (i.e., lists of undiscriminating non-content words, like a, an, the, and so on) are also useful.

WordNet (Miller [1995]) is an example of a knowledge source and is considered a standard in English WSD (Navigli [2009]). This resource can be described as a computational lexicon of English language that organizes concepts as sets of synonyms (synsets). For each synset, it provides a textual definition and possibly some usage examples of the synset. It also provides lexical and semantic relations such as antonymy, hypernymy, hyponymy.

2.2.3 Context Representation

One of the challenges in NLP is that text data is mainly unstructured. To address this issue, many preprocessing methods can be applied to transform text in a structured format. For instance, tokenization (split a text in a set of tokens, usually words), part-of-speech tagging (assign grammatical categories tags to each word, e.g., verb and noun), and lemmatization (reduct morphological variants to their base form, e.g., were $\rightarrow$ be, men $\rightarrow$ man) are commonly applied preprocessing methods. As result of the preprocessing, each word can be represented by a vector of features (e.g., [*token, lemma, part-of-speech tag, frequency, ...*]) or in other structures like graphs or trees. Another alternative is to use distributed representations of words, such as Skip-gram (Mikolov et al. [2013]) or GloVe (Pennington et al. [2014]).

The features used to represent a context can be distinguished into four groups:

- Local: features of a small number of words around the target word.
- Topical: features to represent the general topic of a text
- Syntactic: syntactic relations between the words in the context (usually within the sentence)
- Semantic: semantical information, like senses of words in the context or previously assigned senses.

Another issue is the definition of how much of the document should be considered as context. It can be the sentence where the target words occur or a window of sentences or words surrounding the target word. The window size is usually a hyperparameter of the WSD algorithms.

2.2.4 Classification Method

Classification methods refer to the algorithms used to select the word sense. As mentioned, WSD solutions can be grouped into two main categories, supervised (or semi-supervised) and knowledge-based. It Makes Sense (Zhong and Ng [2010]), which makes use of Support Vector Machines models for word experts and Bidirectional Long Short Term Memory network used in Raganato et al. [2017b] are examples of supervised systems. While Lesk algorithm (Lesk [1986]), which select the sense whose definition has the higher overlap with target word context, is an example of knowledge-based solution.

2.2.5 Evaluation

The evaluation of WSD systems can be intrinsic or extrinsic. The intrinsic evaluation, adopted in this work, aims to evaluate the task itself, by measuring how effective is the WSD solution. An extrinsic evaluation measure how the WSD system impact the downstream tasks. For instance, Peters et al. [2018] evaluates the impact of its method in six other NLP tasks.

Comparisons between WSD systems are not an easy task, some of the difficulty arises from the different sense inventories used to label the sense annotated corpora, or in some cases different versions of the same sense inventory (e.g., WordNet). Raganato et al. [2017b] addressed this issue creating evaluation framework for WSD, which provide normalized versions of fine-grained datasets of Senseval 2 (Edmonds and Cotton [2001]), Senseval 3 (Snyder and Palmer [2004]), SemEval 2007 (Pradhan et al. [2007]), SemEval 2013 (Navigli et al. [2013]), SemEval 2015 (Moro and Navigli [2015]), making easier to compare the intrinsic results of WSD systems.

2.3 Related Work

In this section, we first present different works on knowledge-based systems, with emphasis on Lesk algorithm and its variations. Then we describe an word expert supervised approach which achieves the state of art results in WSD, and a supervised model that address the need of word experts in supervised models providing a more flexible solution to facilitate its usage for downstream tasks.

Lesk algorithm (Lesk [1986]) it is a well-studied knowledge-based algorithm and had various extensions proposed since its introduction. It is an intuitive approach that mimics a human trying to discover the appropriate meaning for a word, whose sense

it don't know, by comparing sense candidates gloss with the word context to select the most appropriate sense for that specific context. The original proposition aimed to disambiguate all words in short sentences at the same time. In this algorithm the gloss of all open class words in the sentence are compared using a overlap measure, which count how many words the sense candidates gloss have in common. The senses combination with largest overlap are assigned to the respective words. Lesk [1986] demonstrated his algorithm for the words "pine cone", using senses retrieved from Oxford Advanced Learner's Dictionary:

Pine senses:

1. "kind of **evergreen tree** with needle-shaped leaves ..."
2. "waste away through sorrow or illness ..."

Cone senses:

1. "solid body which narrows to a point ..."
2. "something of this shape whether solid or hollow ..."
3. "fruit of certain **evergreen trees** ..."

Each sense of the word pine is compared with each sense of the word cone and it is found that the words evergreen tree occurs in one sense each of the two words. These two senses are then assigned as the most appropriate senses when the words pine and cone are used together.

Banerjee and Pedersen [2002] proposed an adapted version of Lesk algorithm. The original algorithm was improved by exploiting the glosses of explicit related words in WordNet (e.g., hypernymy, meronymy, pertainymy, etc.) and a measure of *extended gloss overlap*, which introduces n-grams overlaps that gives more weight to longer overlaps, it is defined as $\sum a_i^2$ where a_i is the length of overlap i. This Lesk algorithm version exploit the related words by extending the gloss comparisons between senses to their hyponyms and hypernyms, where each possible pair of glosses is compared and the highest overlap is considered, the Figure 2.1 show an example of sense comparison. This algorithm was able to outperform the original algorithm, but with a drawback of increasing the computational cost.

One of the issues with the Lesk algorithm was the number of comparisons needed to assign the senses. To address this issue a simplified version of Lesk algorithm was

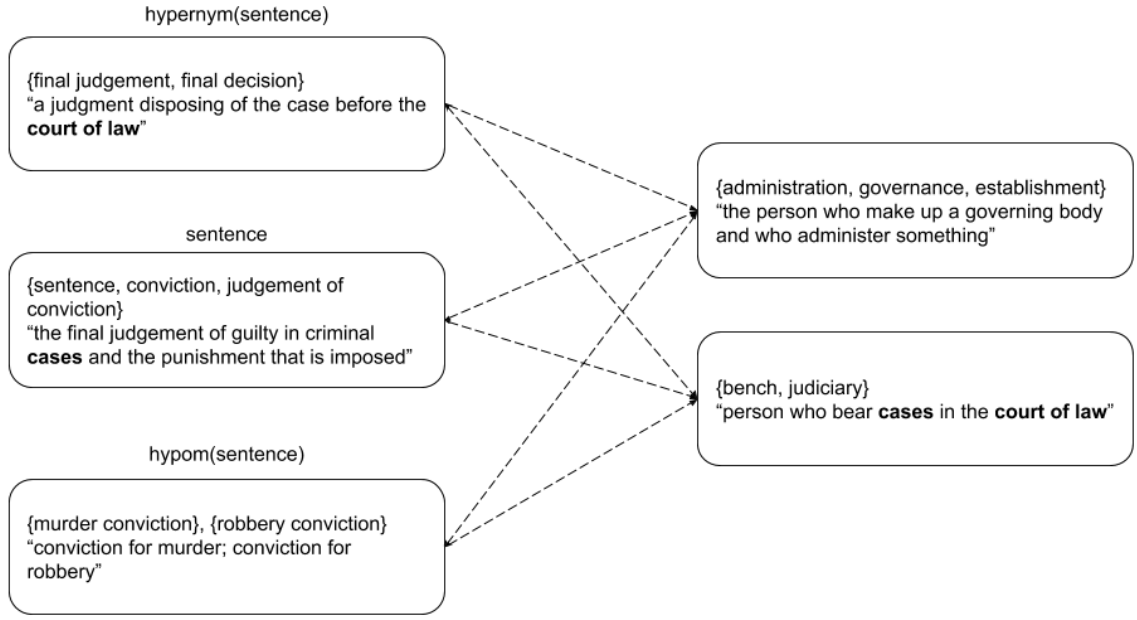


Figure 2.1: Scheme for gloss comparison. Each box denotes a sense, with defining lemmas and the gloss in quotation marks. Dotted lines represent the various gloss-gloss comparisons and the overlaps are showed in bold. Source Banerjee and Pedersen [2002].

proposed by Kilgarriff and Rosenzweig [2000]. In this new algorithm, the overlap measure is calculated between target word context and sense candidate gloss. Another difference of this approach is that each target word is addressed separately and the previously assigned senses are not considered. This simple modification not only diminish the number of gloss comparisons, it improved the resulting disambiguation.

In Basile et al. [2014] the simplified Lesk algorithm was *enhanced*. This algorithm uses cosine similarity between the gloss and context representation to replace the gloss-context overlap measure. The gloss and context are represented by the sum of representations of their words in a word embeddings model. It also used a gloss expansion heuristic using BabelNet (Navigli and Ponzetto [2010]), which is their sense inventory. Similar to Banerjee and Pedersen [2002] the sense candidates relations are also exploited, but in this algorithm they are used to extend sense candidate's gloss, wich is expanded by concatenating it with the related words gloss (e.g., synonyms, hyponyms, hypernyms, etc.) except the antonyms. In order to give more importance to terms occurring in the sense candidates gloss, words in the expanded gloss are weighed taking into account both the distance (number of edges on the knowledge graph) between the sense candidate and the related senses and term frequency. This algorithm

was able to outperform all the participants in SemEval-2013 (Navigli et al. [2013]) on multilingual Word Sense Disambiguation, for both English and Italian.

Unlike the previous similarity-based methods, solutions based on clustering algorithms explore the semantic similarity in a different way. Schütze [1998] split word sense disambiguation into two subtasks: word sense discrimination, which consists in determining if two of occurrences of the same word uses the same sense, while sense labeling consists in intensity which sense was used. Schutze explores words co-occurrences matrix, to create word vectors whose dimensionality is reduced using Singular Value Decomposition (SVD) to create word vectors. Then the cosine similarity between context vectors, given by the mean of word vectors in the context. A cluster is represented by its centroid, calculated by the average of its elements. Other approaches focused on clustering fine-grained word senses into coarse-grained ones to reduce the polysemy degree of ambiguous words (Agirre and de Lacalle [2003]; Navigli [2006]).

Different from the above mentioned methods, supervised solutions usually resort to word expert models, treating each word as a different classification problem. It Makes Sense (IMS, Zhong and Ng [2010]) is an example of supervised model based on Support Vector Machine (SVM) classifiers over a set of conventional WSD features (surrounding words, PoS tags of surroundings words, and local collocations) to create a classifier for each word type. This work was extended by Iacobacci et al. [2016], who studied different embeddings and combinations of features for IMS and is the current state-of-art in fine-grained WSD benchmarks (Raganato et al. [2017b]).

Recently methods based on neural networks are being explored to remove the necessity of word experts common in supervised solutions. Raganato et al. [2017a], for example, studied the use of sequence learning models aiming to remove the need to train word expert models for every word type. Its use of bidirectional Long Short Term Memory (LSTM) networks resulted in a more versatile model with a small loss in performance in comparison with IMS. Sequence learning neural networks can explore the sequential and syntactic information present of the text that would be lost using average or concatenation of word vectors. The work of Yuan et al. [2016] is another example of use of LSTM for WSD in a semi-supervised setting, where a large corpora of text is first used to train a language model which later is used to train a predictive model.

Another approach to WSD relates to the use of sense embeddings. In most word embeddings a word has a single vector as representation failing to model words polysemy. Chen et al. [2014] presented a unified model for word sense representation and disambiguation. In their work, the embeddings dictionary is expanded to include representations of word senses, which are initialized with the average of continuous vector

representations of words in senses gloss. The disambiguation is performed through cosine similarity between the sense and context representation. Based on the disambiguation result a raw text corpora is expanded to include a sense token after each open class word, then the embeddings model are retrained using the expanded text corpora. This method showed good results in domain-specific WSD.

Recently, a new distributed word representation was introduced by Peters et al. [2018] that models both complex characteristics of word use (e.g., syntax and semantics), and how these uses vary across linguistic contexts (i.e., to model polysemy). In their work, word vectors are learned functions of the internal states of a deep bidirectional language model, which is pre-trained on a large text corpus. These representations were added to existing models and significantly improved the state-of-art across six NLP problems, including question answering, textual entailment, and sentiment analysis.

Chapter 3

Proposed Algorithm

In this chapter, we discuss the methods and guidelines used to create our algorithm and its components. The all-words WSD task can be described as the task of identifying the appropriate word sense for all words in a text, except non-content words (e.g., article, preposition). This challenging task can be addressed by retrieving all possible senses a word can assume and then select the sense that better fit for each word in a text. The selection of the appropriate word sense is usually grounded on different sources of information, with the most basic one being a textual description of the word sense. Information about frequency of word senses occurrence, sentences with examples of the word senses usage, relationships between words (e.g., car *is an* automobile), and the context of the word can also be used features to identify the appropriate sense.

The all-words WSD task has two requirements, a text to disambiguate and a sense inventory containing word senses with their respective gloss. Sense inventories play an important role in the WSD and should be selected based on the intended application. The choice of sense inventory must consider if the texts are formal or informal, for example, texts from micro-blogs often have a colloquial language with abbreviations and acronyms usually not present in formal dictionaries. If a word does not pertain the sense inventory vocabulary it cannot be disambiguated, typos, neologism, and foreign languages words are common examples of words outside of vocabulary.

Another import aspect in the sense inventory selection is the desired granularity of word senses. Fine-grained WSD distinguish subtle differences in the word senses, such as 'knife' having different word senses when used as weapon or kitchen tool, while coarse-grained WSD would only attribute 'a blade for cutting' as word sense. In this work, we address the fine-grained all-words WSD task and evaluate the performance of our algorithm in texts from news articles, therefore requiring a formal sense inventory.

External knowledge sources are fundamental to WSD solutions, they provide

useful information to assist the selection of the appropriate word senses. There are various resources that can be explored as knowledge sources, such as ontologies which provide information about word relationships and is often explored by graph based solutions (Moro et al. [2014]). Raw text corpora can also be explored by WSD systems to train distributed word representation models, while sense-annotated corpus can be used to either train supervised approaches or provide information about occurrence frequency of each annotated sense.

Although each knowledge source provides valuable features to WSD solutions, their availability cannot be assured in every scenario. A knowledge source can be unavailable by either being incomplete or missing. The incompleteness relates to the knowledge source not having information about valid words of the input text. For example, in domain-specific applications like physics, words from this particular domain can be missing in a general purpose ontology, hindering WSD approaches that rely on this information. A knowledge source can be missing when a WSD solution is applied in a different domain or language than originally proposed. For instance, a Brazilian version of WordNet (Dias-da-Silva et al. [2008]) does not provide sense frequency information, which is an important feature to Knowledge-Based solutions distinguish between common and rare word senses. This work addresses the problem of missing knowledge sources, providing a flexible compositional solution based on three information sources, capable of performing WSD when some or all knowledge sources are unavailable.

In the following sections we present the proposed algorithm and its three components: sense frequency, gloss extension, and usage examples. We also describe the semantic distance measure used by CLA and the distributed word representation models used to calculate this measure.

3.1 Compositional Lesk Algorithm

Here we present our proposed algorithm, which we call Compositional Lesk Algorithm (CLA). Our algorithm is a context-gloss comparison method that compare gloss and context using a semantic distance measure and selects the sense candidate semantically closer to the context. Methods based on context-gloss comparisons are intuitive in the sense that they behave similarly to a human looking in a dictionary for the meaning of a word, comparing the word senses with the context and then selecting the sense that better fit in that context. CLA explores the intuition that the appropriate word sense should be semantically closer to the context that other sense candidates therefore

using a semantic distance measure as comparison method.

One of the challenges in WSD is to define what is context and how to represent it. In WordNet, linguistic context is described as "discourse that surrounds a language unit and helps to determine its interpretation", but context can also extrapolate linguistics. For instance, consider geographical context, some words can have different meanings in different regions of the same country or an object be known by different names. The communication channel can also be considered part of the context. For example, the word *shoot* usually have different meanings in sports and criminal pages, in another hand, in comedy pages words or phrases can be purposely ambiguous for the sake of a joke. These examples illustrate some of the challenges of general purpose WSD.

In WSD literature, context features are limited to linguistic context and grouped in four categories: local, topic, syntactic, and semantic. Originally, Lesk [1986] proposed the usage of neighbor words to represent the context, which is a form of local context. Our algorithm also makes use of local context by selecting a window of sentences around the target word and represent each word in the window using distributed word representations which give both syntactic and semantic information about words. The word sense gloss uses the same representation, with the difference that the gloss is used completely because it usually contains only one sentence.

The components of our algorithm are each based on a different information, this design aims to make CLA flexible to deal with scenarios of information scarcity, where some or all components are unavailable. Some of the challenges in WSD came from the resources used to solve the task, be it for availability, completeness or quality. From the three CLA components two relates to quality of the word senses glosses which can be either very short, sometimes having only one word, or multiple senses having similar descriptions. The extended gloss component aims to ease this problem by enriching a word sense's gloss using gloss of related words. The usage examples component also addresses the gloss quality problem enhancing the gloss using sentences with usage examples of the word senses. The sense frequency component addresses the issue of rare word senses and is used to weight the semantic distance between context and gloss by the occurrence probability of a word sense. The Algorithm 1 demonstrates how the components are combined when available.

The Figure 3.1 presents an example of CLA disambiguating the word *shoot* in 'she made the game winning shoot'. In this example we show some of the word sense for the word shoot and demonstrates the information obtained from the components for the word sense *score*. Extended gloss and usage examples provide short sentences to improve the gloss quality while sense frequency provides the probability of *shoot* assume *score* as word sense. The sense score output is used as to select the most

Algorithm 1: Compositional Lesk Algorithm

```
Data: Sense Inventory, Text, Embedding Model
Result: selected senses
for content_word in Text do
    context ← get_local_context(content_word, Text);
    sense_candidates ← sense_inventory.get_senses(content_word);
    for candidate in sense_candidates do
        min_distance ← ∞;
        selected_sense ← NULL;
        gloss = candidate.gloss();
        if ontology is available then
            gloss = expand_with_related_words(gloss);
        end
        if usage_examples is available then
            gloss = expand_with_usage_examples(gloss);
        end
        distance ← WMD(context, gloss);
        if sense_frequency is available then
            weight ← 1 - get_probability(sense_candidate);
            distance ← weight * distance
        end
        if distance < min_distance then
            min_distance ← distance;
            selected_sense ← candidate;
        end
    end
    set_sense(content_word, selected_sense);
end
```

appropriate sense for this context. This a good example of how a sense candidate can have a very short description and how the sense frequency can be misleading if the domain used to count the sense occurrence be different from the application domain.

3.2 Components

In this section we further describe each CLA components, detailing the motivation them and how they are used in the proposed algorithm.

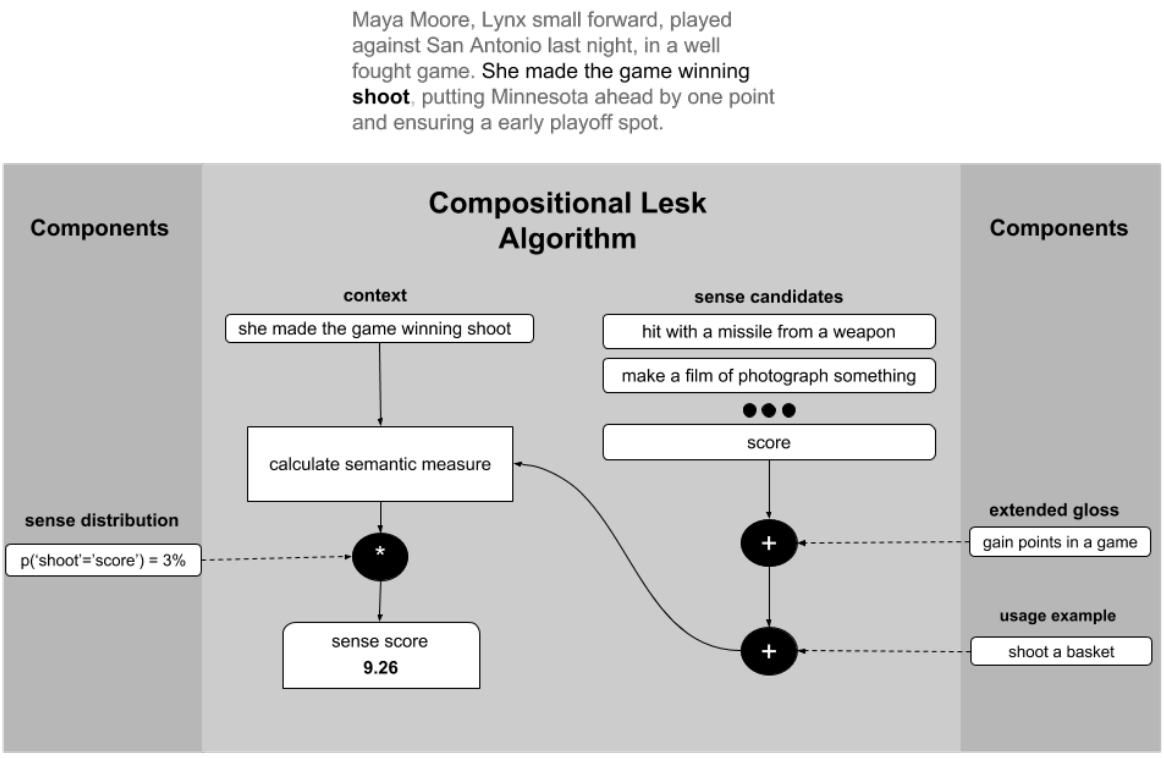


Figure 3.1: Compositional Lesk algorithm disambiguating the word *shoot* using all components

3.2.1 Sense Frequency

Sense frequency refers to the number of times a word sense occurs in an annotated corpus. For instance, WordNet (Miller [1995]) provides information about sense frequency based on the SemCor corpus (Miller et al. [1993]). This information is commonly used in knowledge-based approaches, and all baselines presented in Section 4.2 rely on this information. Similar to Basile et al. [2014], we use Laplace smoothing to calculate the probability of a word assume each of its sense candidate, with the difference that we study the smoothing parameter impact on the development set results.

Laplace smoothing, also known as additive smoothing, is a method smooth the the probability distribution of a sample by adding α examples of each class in the sample. The Laplace smoothing equation is defined as:

$$p_s = \frac{freq_s + \alpha}{N + d * \alpha}$$

Where p_s is the target word's probability of occur with the sense s, $freq_s$ is the

number of times the target word appears in the corpus with sense s, d is the number of sense candidates and N is the target word's frequency (sum of all sense frequencies). α is the smoothing parameter, it leverages how much weight will be given to observations in the sample. If α is zero the observations will have maximum weight, but as α grows larger the probability distribution gets closer to an uniform distribution, minimizing observations weight.

In CLA the sense probability is used to leverage the semantic distance. Aiming to weigh how close a word sense and context are with the word sense probability, we multiply the semantic distance with the complement of sense probability ($(1 - p_s) * distance$). The complement of sense probability is used since we intend to shorten the distance of frequent senses more than seldom ones. A possible drawback of using sense frequency is the strong bias towards the most frequent sense of both supervised and knowledge-based solutions. For instance, Babelfy (Moro et al. [2014]) includes a confidence threshold in order to decide whether to disambiguate or back-off to the most frequent sense, backs-off in 63% of all instances. To ease this bias towards the most frequent sense we evaluate different values of α in the Laplace smoothing to leverage the sense frequency importance.

3.2.2 Extended Gloss

A difficulty encountered by methods that rely on senses gloss to perform WSD is the gloss quality. In some cases, the gloss can be so small that the method is not able to draw conclusions of it, or two senses having very similar definitions. For instance, in WordNet, the noun *play* most frequent senses are:

1. A dramatic work intended for performance by actors on a stage.

2. A theatrical performance of a drama.

Although they have very similar definitions, the first sense refers to any dramatic composition itself (e.g., script, scene or act), while the second refers to the dramatic composition performance. Without more information, it's hard to distinguish between both these senses.

One solution used both by Banerjee and Pedersen [2002] and Basile et al. [2014] is to explore related words in an ontology to improve the disambiguation, although using different approaches in how to use these relations, both works show the potential for improving the WSD quality using a external knowledge source. We also explore these relations by concatenating the hypernyms and hyponyms gloss to sense candidate

definition. While CLA can make use of this information if it's available, it can also work independently of its availability. In Chapter 4 we evaluate the extended gloss impact on WSD efficacy.

3.2.3 Usage Examples

Some sense inventories, such as WordNet, provides some sentences with usage examples of some senses. Although it's not available for every sense, we also explored this resource to improve the gloss quality. In particular, we propose two ways to enhance the gloss, first is replace sense definition by its usage examples when it is available, based on the idea that the target word context should be more similar to the sense usage example than its definition. The second is concatenate the sense gloss with its examples (more than one example can be available), this is approach is more similar to the extended gloss in which we intend to enhance the gloss by adding more information to it.

We evaluate both methods of adding sense usage examples as well as combining it with the gloss extension described in the last section. When combined, the gloss extension follows the setup described in this section, in other words, when using replacement as method of introducing usage examples we combining with the hypernyms and hyponyms usage examples are used instead of its definition, while the other method concatenates all available resources (i.e., gloss and usage examples of the sense candidate and related words).

3.3 Semantic Distance

A key element in context-gloss comparison methods is the comparison mechanism. The comparison between gloss and context is based on the intuition that the appropriate sense for a word should be similar to the context where it word appears. A challenge of similarity measures is capture semantic information between two texts. For instance, Lesk [1986] originally proposed an overlap measure which counted words co-occurring in both gloss and context, an important drawback of this solution is the incapability identifying texts that are semantically similar, but written with completely different words. In CLA we use Word Mover's Distance (WMD, Kusner et al. [2015]), a hyperparameter free and interpretable distance between two texts that captures semantic information.

WMD is built over semantically meaningful word representations. It measures the dissimilarity between two texts as the minimum distance embedded words from a document needs to "travel" in the semantic space to reach the embedded words of

another document. An illustration of WMD is presented in Figure 3.2 that shows an example of semantically similar sentences that does not have any words in common. This semantic measure provides to CLA a straight-forward comparison method allowing CLA to explore syntactic and semantic features encoded in the word embeddings.

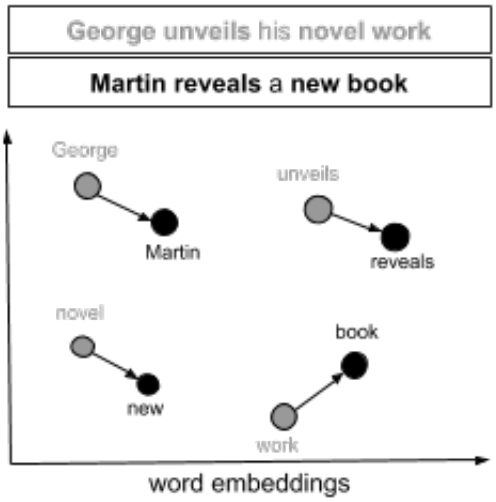


Figure 3.2: Arrows represent the distance an embedded word needs to "travel" to reach the words of the other document. Note that non-content words are disregarded.

3.3.1 Embeddings

Word embeddings, sometimes called distributed semantic model or distributed word representations, is a continuous vector representation of a word or the context where it appears. Word embeddings models are widely used in many NLP tasks, inclusive in WSD (Iacobacci et al. [2016]; Raganato et al. [2017a]). In this work, we also explore distributed word representations through WMD, which relies on these representations to calculate the semantic distance. Therefore, we also evaluated two different embeddings model GloVe and Skip-gram.

GloVe (Pennington et al. [2014]) is a count-based method, which first constructs a large co-occurrences count matrix (i.e., number of times two words occur in the same context). Then, it uses a matrix factorization method to learns the word representations by reducing the dimensionality of a co-occurrence count matrix. In contrast, Skip-gram (Mikolov et al. [2013]) is a method based on artificial neural networks, which learns the word representations by solving the task of given a word predict its context (i.e., its neighbor words). We evaluate all the above-mentioned components of our proposed method using each of these embeddings to investigate these models impact in our approach. The results of these evaluations are reported in the next chapter.

Chapter 4

Results and Discussion

In this chapter, we present the datasets used for development and evaluation, discuss our evaluation procedure, and report the results of our proposed algorithm, along with other specific results of this work. In particular, our experiments aim to answer the following research questions:

RQ1: Which CLA component is more important? (Answered in Section: 4.4.1)

RQ2: How the components interact with one another? (Answered in Section: 4.4.2)

RQ3: How different embeddings impact CLA? (Answered in Section: 4.4.3)

RQ4: How the different components perform when compared with other knowledge based and supervised approaches? (Answered in Section: 4.4.4)

4.1 Datasets

SemEval, short for semantic evaluation, is an ongoing series of workshops with a focus on computational semantic analysis that provided several corpora of manually annotated data for a wide range NLP tasks. It was originally called Senseval and was renamed after three editions of the workshop. This workshop series addresses tasks such as Textual Similarity, Question Answering, Sentiment Analysis, Semantic Parsing, and WSD.

One of the challenges in semantic evaluation tasks is the everchanging language. New words are created or new senses are assigned to existing words (e.g., 'tweet' referring to postings in a social media website). Therefore, language resources are constantly updated to match languages evolution. The problem is that the manual annotations are not reviewed along with language resources. Raganato et al. [2017b] addressed this issue presenting a standardized version of five Senseval/SemEval manually annotated

datasets for WSD: **Senseval-2** (SE2, Edmonds and Cotton [2001]), **Senseval-3** (SE3, Snyder and Palmer [2004]), **SemEval-2007** (SE7, Pradhan et al. [2007]), **SemEval-2013** (SE13, Navigli et al. [2013]), **SemEval-2015** (SE15, Moro and Navigli [2015]). Following Raganato et al. [2017a], we used SemEval-2007 dataset as development set while the others were reserved for evaluation.

Table 4.1: Statistics of the fine-grained WSD datasets.

	#Doc	#Sents	#Tokens	#Annotations	#Senses	#Words	Ambiguity
SE2	3	242	5766	2282	1335	1093	5.4
SE3	3	352	5541	1850	1167	977	6.8
SE7	3	135	3201	455	375	330	8.5
SE13	13	306	8391	1644	827	751	4.9
SE15	4	138	2604	1022	659	512	5.5
Total	26	1173	25503	7253	4363	3663	5.8

The Table 4.1 present some statistics about the datasets used in this work. In the table, #Annotations refers to the number sense-annotated words in the corpus, #Senses refers to the number of different senses and #Words refer to amount of unique annotated words. Is easy to notice that in every dataset there are more sense types than word types, therefore assuring that there are word types annotated with different senses. And the number of sense annotations varies across datasets, ranging from 455 annotations in SemEval-2007 to 2,282 in Senseval-2. Ambiguity refers to the ambiguity level of each dataset, calculated as the division between the total number sense candidates and the number of sense annotations [Raganato et al. [2017b]], the ambiguity level can indicate how difficult is a given dataset.

Table 4.2: Number of annotated words per Part-of-Speech.

	#Adv.	#Verbs	#Adj.	#Nouns	Total
SE2	254	517	445	1066	2282
SE3	12	588	350	900	1850
SE7	0	296	0	159	455
SE13	0	0	0	1644	1644
SE15	80	251	160	531	1022
Total	346	1652	955	4300	7253
Ambiguity	3.1	10.4	3.8	4.8	5.8

The Table 4.2 presents information about the number of annotated words per PoS (Part-of-Speech) tag for each dataset. Here we can see that SemEval-2007 dataset do not have annotations for adjectives and adverbs and SemEval-2013 only has annotations for nouns, and even datasets with annotations for all PoS tags have an imbalanced amount of annotations for each of them. Considering all datasets nouns annotations

represent 59.3% of all sense annotations, verbs represents 22.8%, adjectives 13.1% and adverbs 4.8%. This table also presents the ambiguity level per PoS tag in the concatenation of all datasets, and indicates that verbs are the most difficult words to disambiguate.

4.2 Baselines

We considered the following methods in order to provide a baseline comparison:

1. **Most Frequent Sense (MFS)**: This baseline consists in choosing the sense a word was most used with in an annotated corpus. For example if the word *knife* was used 4 times with the sense 'kitchen tool' and 2 times with sense 'weapon' this approach will aways associate 'kitchen tool' as word sense for *knife*. For instance, the list of senses for each word provided by WordNet (Miller [1995]) are ordered based on the sense occurrence frequency in SemCor corpus (Miller et al. [1993]). In other words, this baseline chooses the most frequent sense in a sense-tagged dataset and consequently only one sense is assigned for word type.

2. **Enhanced Lesk Algorithm ($\mathbf{Lesk_{ext}}$+emb)**: this approach was proposed by Basile et al. [2014], is a context-gloss method where gloss and context are represented by the mean of their word embeddings and cosine similarity is used as comparison mechanism. It also extends the sense candidates gloss using related words in an ontology. This algorithm was first proposed using Distributed Semantic Models and was extended to use Skip-gram embeddings by Raganato et al. [2017b].

3. **Babelfy**: a state-of-art knowledge-based WSD solution. Its a unified approach for both Entity Linking and WSD based on the identification of candidate meanings in a semantic network coupled with a densest subgraph heuristic which selects high-coherence semantic interpretations. Babelfy (Moro et al. [2014]) requires the availability of a wide-coverage semantic network which encodes structural and lexical information both of an encyclopedic and of a lexicographic kind.

4. **IMS**: originally proposed by (Zhong and Ng [2010]), IMS is a supervised method that uses a Support Vector Machine (SVM) classifier over a set of conventional WSD features such as surrounding words, PoS tags, and local collocations to train word experts for each word type in the training corpus. More recently Iacobacci et al. [2016] compared different ways of integrating word embeddings as a feature in WSD using IMS framework improving its results.

5. **Context2Vec**: proposed by Melamud et al. [2016], this is a supervised approach based on sentence representations that can be described in three steps: (1) Train a bidirectional LSTM on an unlabeled corpus. (2) Use the training corpus to learn a context vector for each sense annotation. (3) Select the sense annotation whose context vector is closer to the target word's context vector.

6. **BiLSTM**: the supervised approach proposed by Raganato et al. [2017a] uses a bidirectional Long Short Term Memory (LSTM) network to explore the sequential and syntactic information present of the text that would be lost using average or concatenation of word vectors. As certain elements of the sentence might be more discriminative than other in predicting the output label the bidirectional LSTM is coupled with a attention mechanism.

4.3 Setup

In this section we describe the experiments to tune CLA hyper-parameters. The two hyper-parameters in CLA are window size and smoothing parameters. Window size relates to the context representation, this parameter define the number of sentences to be considered to represent the context. The smoothing parameter pertain to Laplace smoothing used in the sense frequency component to calculate the probability of a word assume each sense candidate. As the smoothing parameter tends towards greater values th e probability distribution approximates to the uniform distribution. Therefore, the greater the smoothing parameter less weight is given to sense frequency.

We used SE7 as the development set to evaluated different window sizes, ranging from 0 (only the target word sentence) to 8 (8 sentences at left and 8 at right, with a max of 17 sentences), and the smoothing parameter ranging from 1 to 8 for each combination of components (e.g., extended gloss, usage examples, extended gloss + usage examples). Grid search was utilized to search for the better set of parameters, the result of this search is displayed in Table 4.3.

4.4 Answering the Research Questions

Here we present and discuss our experimental results for different scenarios of information availability and answer the research questions presented in the beginning of this chapter.

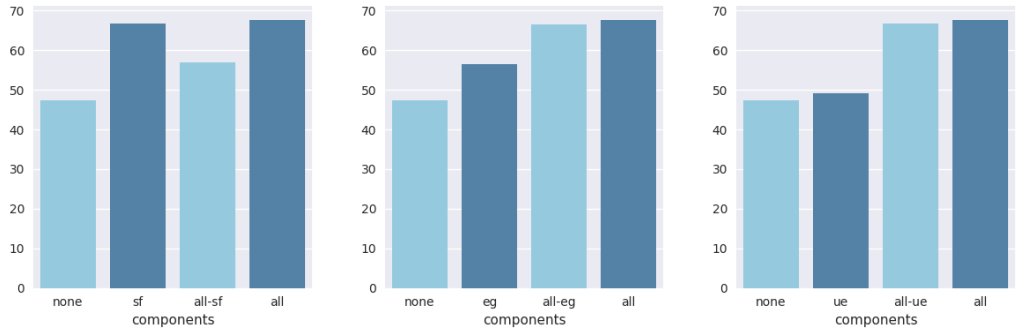
Table 4.3: Best parameters found using grid search in the development dataset (SE7) for each combination of CLA components. Examples refer to Usage Examples component, S. Freq to Sense Frequency, and Ext. Gloss to Extended Gloss component. Window refers to the number of sentences considered as context and α to the value of the smoothing parameter used with the sense frequency component.

Emb.	Examples	S. Freq	Ext. Gloss	Window	α
GloVe	None	yes	yes	0	3
			no	0	1
		no	yes	5	-
			no	0	-
	Replace	yes	yes	0	8
			no	0	2
		no	yes	3	-
			no	0	-
	Concat	yes	yes	0	8
			no	0	1
		no	yes	5	-
			no	0	-
Skip-gram	None	yes	yes	0	3
			no	0	2
		no	yes	8	-
			no	0	-
	Replace	yes	yes	0	4
			no	0	3
		no	yes	8	-
			no	0	-
	Concat	yes	yes	1	8
			no	0	2
		no	yes	8	-
			no	0	-

4.4.1 Component Importance

In this section we answer our first research question **RQ1**, which aims to identify which component contributes more to algorithm overall performance. To answer this question we compared the improvement in effectiveness of adding a component in two different scenarios, first with no external source of knowledge available (i.e., using only the semantic distance between context and gloss), and later with all but the analyzed component in use. In other words, add the component when no other are available and remove the component when all are available. The Figure 4.1 presents the improvement of adding each component/information in both scenarios described.

The addition of sense frequency component presented most improvement, which is not a surprise since MFS is a hard to beat baseline in WSD (Navigli [2009]). Sense



(a) Sense frequency (b) Extended gloss (c) Usage Examples

Figure 4.1: (Colors online) F1 measure increase of adding each component in two scenarios: only the dictionary gloss is available, (left pair); the two other components are available (right pair).

frequency importance relates to the capacity to distinguish between common and rare senses, giving the algorithm ability to leverage the semantic distance with word sense probability. It shows improvements of 19.4 in the F1 score when added in a scenario without any component and 10.7 when all other components were already present.

Gloss extension and usage examples components address the problem of gloss quality. Both components address lack of clarity, completeness and other issues in word senses gloss. The gloss extension component provided 9.2 improvement in the F1 score over the scenario without components and approximately 1 when combined with the other two components, and the usage examples component contributed with 1.8 and 1.1 respectively.

4.4.2 Components Interaction

In this section we address our second research question **RQ2** related to the interactions between component in CLA. The aim of our analysis was discover if a component affects the others and how different combinations of components impact the overall disambiguation performance. To evaluate the interaction between components we simulated eight scenarios of information availability to evaluate each possible combination of components. For every scenario we used grid search to look for the better set of hyper-parameters described in 4.3, the results of this search are displayed in Table 4.3.

The first results related to interaction between components was observed during the parameters search. The Table 4.3 that shows the best values for window size and smoothing parameter for each scenario of information availability, shows that when all components (sense frequency, extended gloss and usage examples) are available

the smoothing parameter tends towards a higher value which is usually smaller when extended gloss or usage examples components are unavailable. A possible explanation for this behavior is that when more information is available less weight is given to the sense frequency. Other aspect observed in parameter search was related to context window size, that was greater when the gloss was enhanced using either usage examples or extended gloss components.

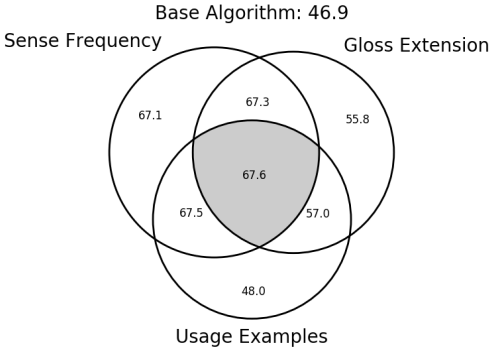
Following the parameter search we evaluated CLA in each scenario of information availability using SE2, SE3, SE13, SE15, and the concatenation of these four datasets. The results of our experiments are presented in form of Venn diagrams, were each set represents a component and the intersections represents scenarios were two or more components are used. The result were no component are available is displayed above the Venn diagram. Figure 4.2 presents the results for SE2, SE3, SE13, SE15 datasets and for the concatenation of all test sets.

Two aspects we first observed were that scenarios with at least one component outperformed the scenario with no component available for every dataset. Also, the scenario with all components available consistently provided better results than any other scenario. Usually the addition of an information helped to improve the algorithm performance, but in some occasions the addition of usage examples had an slight decrease in the F1 score when compared with only using sense frequency or extended gloss.

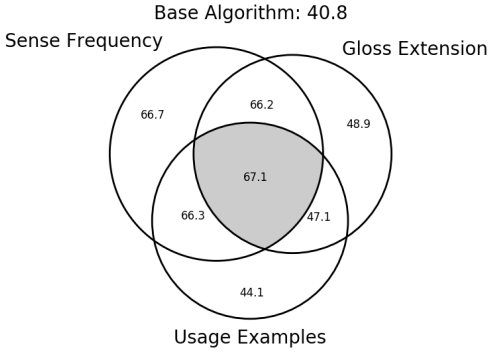
Figure 4.4 presents the results considering the words part-of-speech. One of the most noticeable aspect in this figure is the difference in performance between PoS, while CLA achieves an F1 score of 81.2 in adverbs, in verbs the best result is 51.5. Its interesting to notice that the performance for each PoS was directly related to the ambiguity level presented in Table 4.2. Another curious aspect observed was the change in importance order of components changes for different PoS tags, for adverbs and adjectives usage examples contributes more to the algorithm performance than extended gloss.

4.4.3 Embeddings

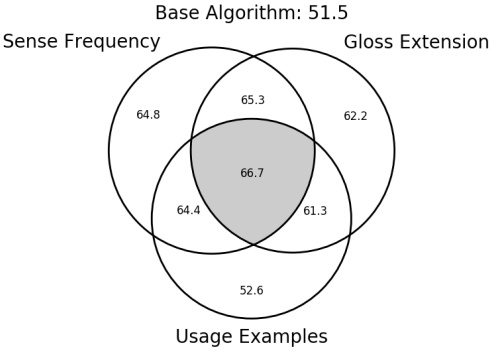
In this section we answer our third research question **RQ3**, which regard the embeddings model impact in our proposed algorithm. To investigate this we evaluated the effectiveness of CLA in the simulated scenario using two well known embeddings models, GloVe and Skip-Gram. The Figure 4.2 presents the experimental results of each scenario using GloVe, and the Figure 4.3 presents the results of using Skip-Gram as embedding model.



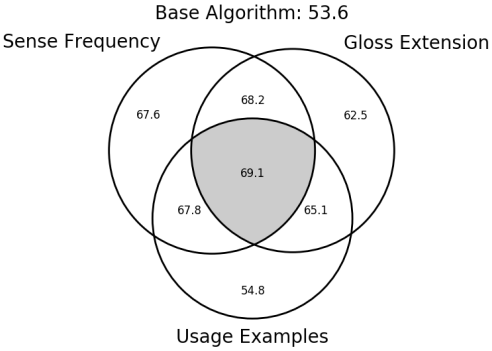
(a) SE2



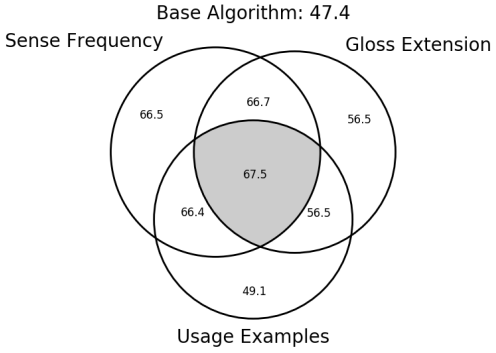
(b) SE3



(c) SE13



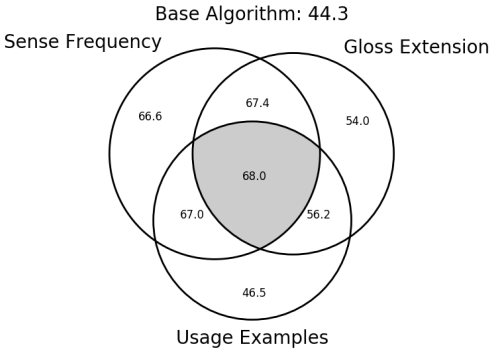
(d) SE15



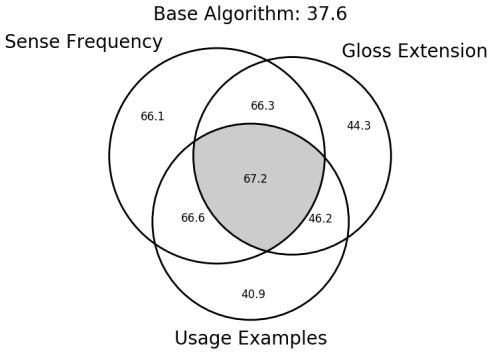
(e) All

Figure 4.2: The F1 score of CLA using GloVe on all evaluation datasets for each possible combination of components. In the Venn diagram, each 'set' represents a component, and intersections represent scenarios where more than one components are available.

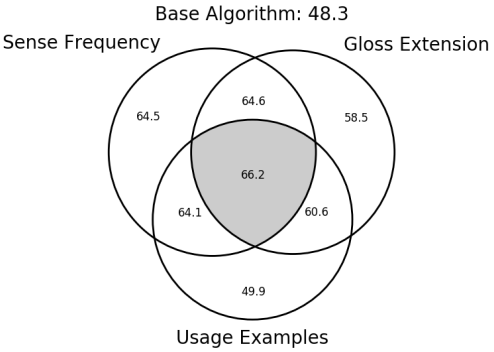
We observed that GloVe consistently outperformed Skip-Gram when there is no component available, this was also observed when only one component was available. Although GloVe performed better in almost every scenario both embeddings performed similarly when all components were available.



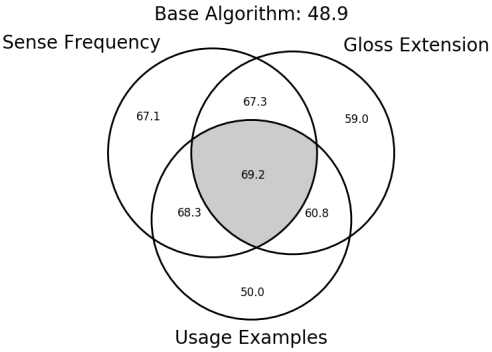
(a) SE2



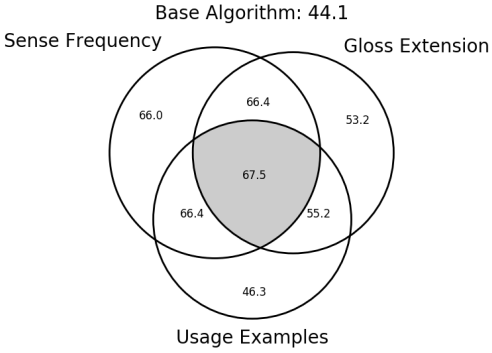
(b) SE3



(c) SE13

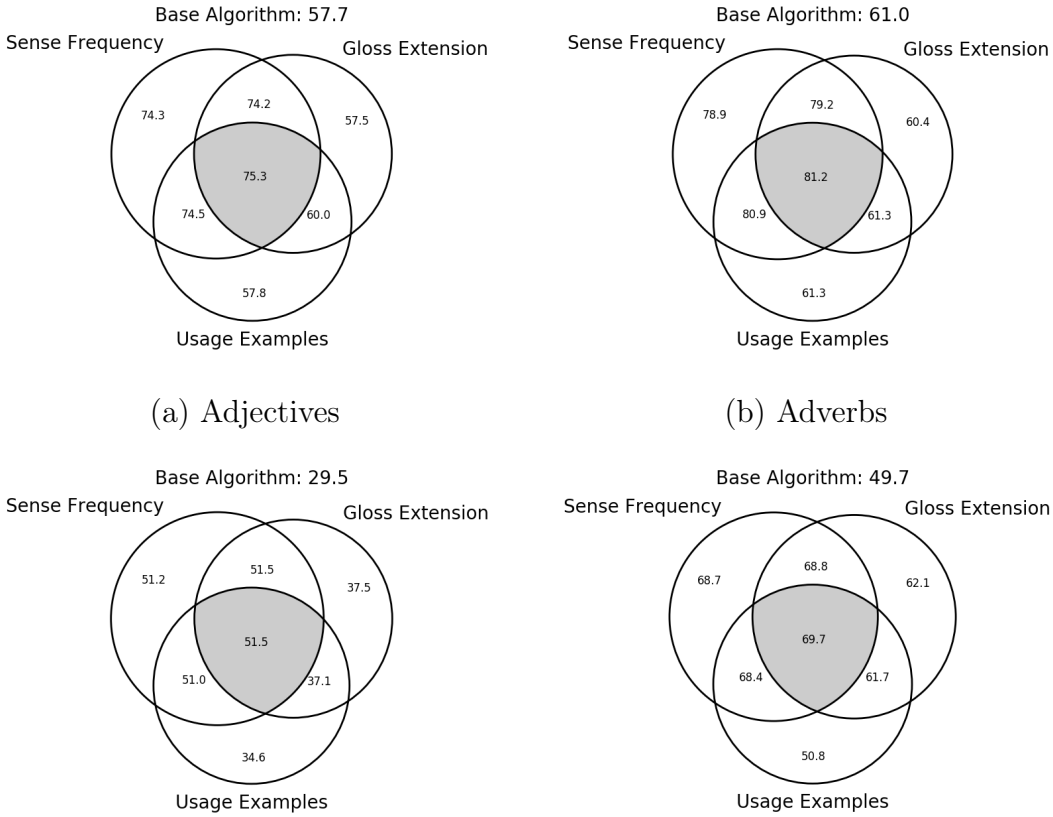


(d) SE15



(e) All

Figure 4.3: The F1 score of CLA using Skip-Gram on all evaluation datasets for each possible combination of components. In the Venn diagram, each 'set' represents a component, and intersections represent scenarios where more than one components are available.



(a) Adjectives

(b) Adverbs

(c) Verbs

(d) Nouns

Figure 4.4: The F1 score of CLA per PoS on the concatenation of all evaluation datasets for each possible combination of components . In the Venn diagram, each 'set' represents a component, and intersections represent scenarios where more than one components are available.

4.4.4 Baselines comparison

Answering our final research question **RQ4**, we compared the results of CLA to other knowledge-based and supervised approaches described in Section 4.2. The Table 4.4 presents baselines and CLA results for each evaluation dataset and their concatenation. This results reflect a scenario where all components/information are available to CLA, since its the same scenario in which baselines were evaluated. In this Table CLA shows comparable results with baselines knowledge-based WSD systems, showing the potential of this approach.

Table 4.4: F-measure for English all-words fine-grained WSD on the test sets

	Algorithm	SE2	SE3	SE13	SE15	All Test
Supervised	IMS+$_{emb}$	**72.2**	**70.4**	65.9	**71.5**	**70.1**
	Context2Vec	71.8	69.1	65.6	71.9	69.6
	BiLSTM	72.0	69.1	**66.9**	71.5	69.9
Supervised	$Lesk_{ext} + emb$	63.0	63.7	66.2	64.6	64.2
	Babelfy	67.0	63.5	66.4	**70.3**	66.4
	MFS	65.6	66.0	63.8	67.1	65.5
	CLA	**68.0**	**67.2**	**66.7**	69.2	**67.5**

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this work we proposed CLA, a flexible compositional system for WSD based on document level semantic distance. This approach was able to perform word sense disambiguation in different scenarios of information availability, inclusive when only the sense inventory is available, scenario in which no supervised solution can be used due to the lack of training data, as well all the baselines presented, due to sense frequency and related words information not being available.

We also evaluated CLA in different information availability scenarios (i.e., different combinations of knowledge sources are available at time), analyzed the component importance, and presented the results of proposed components interaction with each other. We showed that the addition of any component improves result over the base algorithm. In terms of component importance, sense frequency demonstrated the highest improvements in effectiveness, followed by extended gloss and usage examples. In the scenario where all of its components are available (i.e., scenario in which the baselines were evaluated), CLA has results comparable with state-of-art knowledge-based systems.

As the semantic distance used in CLA is based on distributed word representations, we also evaluated the performance of each components combination using two different pre-trained word embeddings models: GloVe, a count-based method based on matrix factorization and Skip-gram, which learns the word representations using artificial neural networks. Although they performed differently in some scenarios, both models performed similarly when all components are available.

5.2 Future Work

In this work we proposed a flexible compositional model for word sense disambiguation to address different scenarios of information availability. We evaluated the proposed model on the standardized fine grained datasets, however we lacked standardized coarse-grained datasets. Since WSD algorithms based on context-gloss comparison struggle to disambiguate between similar senses, we believe it would perform better distinguishing between coarse grained-senses, and therefore we intend to evaluate CLA in coarse-grained. A solution for the lack of standardized data would be the usage of Raganato et al. [2017b] framework to preprocess coarse-grained datasets like task 7 of SemEval 2007 (Navigli et al. [2007]).

Although being evaluated exclusively in english language, CLA is language independent, and its base algorithm only requires a sense inventory and a text corpora (to train the word embeddings model) or a pre-trained distributed word representation model. The choice of english language was due to two main reasons: comparison with results previously reported in WSD literature and the scarcity of multi-language WSD datasets, and in some cases the existing datasets are annotated with older versions of sense inventories (WordNet), which is a problem since the language is ever changing, as new words are added or new senses associated with existing words. Despite the difficulties, we plan to evaluate CLA in other languages.

Distributed word representations although are very useful to calculate the semantic distance used in CLA, they usually don't model polysemy, therefor vector of a word in the embeddings model usually represents the most common sense of that word type in the embeddings training corpora. In the last years various attempts to create polysemy-aware word representations has been made (Chen et al. [2014]; Peters et al. [2018]) and we plan to evaluate the impact a this embeddings have in the proposed algorithm.

Bibliography

Agirre, E. and de Lacalle, O. L. (2003). Clustering wordnet word senses. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, *Recent Advances in Natural Language Processing III*, volume 260 of *Current Issues in Linguistic Theory (CILT)*, pages 121--130. John Benjamins, Amsterdam/Philadelphia.

Agirre, E., de Lacalle, O. L., and Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57--84.

Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, pages 136--145.

Basile, P., Caputo, A., and Semeraro, G. (2014). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 1591--1600.

Bovi, C. D., Telesca, L., and Navigli, R. (2015). Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529--543.

Carpuat, M. and Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61--72. Association for Computational Linguistics.

Chen, X., Liu, Z., and Sun, M. (2014). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1025--1035. Association for Computational Linguistics.

de Lacalle, O. L. and Agirre, E. (2015). A methodology for word sense disambiguation at 90% based on large-scale crowdsourcing. In *Proceedings of the Fourth Joint*

Conference on Lexical and Computational Semantics, pages 61--70. The *SEM 2015 Organizing Committee.

Dias-da-Silva, B. C., Felippo, A. D., and das Graças Volpe Nunes, M. (2008). The automatic mapping of princeton wordnet lexical-conceptual relations onto the brazilian portuguese wordnet database. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Edmonds, P. and Cotton, S. (2001). SENSEVAL-2: overview. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1--5. Association for Computational Linguistics.

Flekova, L. and Gurevych, I. (2016). Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting on Association for Computational*. Association for Computer Linguistics.

Hornby, A. S. and Crowther, J. (1963). *Oxford advanced learner's dictionary*. Oxford University Press.

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2015). Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53th Annual Meeting on Association for Computational*, pages 95--105. Association for Computer Linguistics.

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computer Linguistics.

Kilgarriff, A. and Rosenzweig, J. (2000). Framework and results for english SENSEVAL. *Computers and the Humanities*, 34(1-2):15--48.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 957--966. JMLR.org.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24--26. Association for Computing Machinery.

Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51--61.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *The Computing Research Repository (CoRR)*, abs/1301.3781.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39--41.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 303--308. Association for Computational Linguistics.

Moro, A. and Navigli, R. (2015). Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 288--297. Association for Computer Linguistics.

Moro, A., Raganato, A., and Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231--244.

Navigli, R. (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of 21st International Conference on Computational Linguistics*.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1--10:69.

Navigli, R., Jurgens, D., and Vannella, D. (2013). Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 222--231. Association for Computer Linguistics.

Navigli, R., Litkowski, K. C., and Hargraves, O. (2007). Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 30--35. Association for Computational Linguistics.

Navigli, R. and Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for*

Computational Linguistics, ACL '10, pages 216--225. Association for Computational Linguistics.

Neale, S., Gomes, L., Agirre, E., de Lacalle, O. L., and Branco, A. (2016). Proceedings of the tenth international conference on language resources and evaluation. In *LREC*. European Language Resources Association (ELRA).

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532--1543. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. volume 1 (Long Papers), pages 2227--2237.

Pradhan, S., Loper, E., Dligach, D., and Palmer, M. (2007). Semeval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87--92. Association for Computer Linguistics.

Raganato, A., Bovi, C. D., and Navigli, R. (2017a). Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156--1167. Association for Computational Linguistics.

Raganato, A., Camacho-Collados, J., and Navigli, R. (2017b). Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 99--110. Association for Computational Linguistics.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97--123.

Snyder, B. and Palmer, M. (2004). The english all-words task. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Association for Computational Linguistics.

Weissenborn, D., Hennig, L., Xu, F., and Uszkoreit, H. (2015). Multi-objective optimization for the joint disambiguation of nouns and named entities. In *Proceedings of the 53th Annual Meeting on Association for Computational*, pages 596--605. Association for Computer Linguistics.

Xiong, D. and Zhang, M. (2014). A sense-based translation model for statistical machine translation. In *Proceedings of the 52th Annual Meeting on Association for Computational*, pages 1459--1469. Association for Computer Linguistics.

Yuan, D., Richardson, J., Doherty, R., Evans, C., and Altendorf, E. (2016). Semi-supervised word sense disambiguation with neural models. In *26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pages 1374--1385. Association for Computational Linguistics.

Zhong, Z. and Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the Association for Computer Linguistics 2010 (System Demonstrations)*, pages 78--83. Association for Computer Linguistics.

Zhong, Z. and Ng, H. T. (2012). Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting on Association for Computational*, pages 273--282. Association for Computer Linguistics.