SAPIENZA
UNIVERSITÀ DI ROMA

Semantic Vector Representations of Word Senses, Concepts and Entities and their Applications in Natural Language Processing

Scuola di Dottorato in Informatica

Dottorato di Ricerca in Informatica – XXX Ciclo

Candidate

Jose Camacho Collados
ID number 1644576

Thesis Advisors

Prof. Roberto Navigli (until 2017)
Prof. Giovanni Stilo (2017 onward)
Prof. Mohammad Taher Pilehvar (co-advisor)

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

October 2017

Thesis defended on 12 February 2018

Semantic Vector Representations of Word Senses, Concepts and Entities and their Applications in Natural Language Processing
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: collados@di.uniroma1.it

Abstract

Representation learning lies at the core of Artificial Intelligence (AI) and Natural Language Processing (NLP). Most recent research has focused on develop representations at the word level. In particular, the representation of words in a vector space has been viewed as one of the most important successes of lexical semantics and NLP in recent years. The generalization power and flexibility of these representations have enabled their integration into a wide variety of text-based applications, where they have proved extremely beneficial. However, these representations are hampered by an important limitation, as they are unable to model different meanings of the same word.

In order to deal with this issue, in this thesis we analyze and develop flexible semantic representations of meanings, i.e. senses, concepts and entities. This finer distinction enables us to model semantic information at a deeper level, which in turn is essential for dealing with ambiguity.

In addition, we view these (vector) representations as a connecting bridge between lexical resources and textual data, encoding knowledge from both sources. We argue that these sense-level representations, similarly to the importance of word embeddings, constitute a first step for seamlessly integrating explicit knowledge into NLP applications, while focusing on the deeper sense level. Its use does not only aim at solving the inherent lexical ambiguity of language, but also represents a first step to the integration of background knowledge into NLP applications. Multilinguality is another key feature of these representations, as we explore the construction language-independent and multilingual techniques that can be applied to arbitrary languages, and also across languages.

We propose simple unsupervised and supervised frameworks which make use of these vector representations for word sense disambiguation, a key application in natural language understanding, and other downstream applications such as text categorization and sentiment analysis. Given the nature of the vectors, we also investigate their effectiveness for improving and enriching knowledge bases, by reducing the sense granularity of their sense inventories and extending them with domain labels, hypernyms and collocations.

Acknowledgements

There are many people that helped throughout the journey of my PhD. Unfortunately, I cannot mention all of them here, although this would have been surely not the same without all these interactions.

First, I would like to thank the Linguistic Computing Laboratory (LCL) from Sapienza University in Rome, led by Prof. Navigli, for giving me the opportunity to start a PhD. Then, I would also like to thank Google for funding my PhD from my second year through the Google Doctoral Fellowship program. It has been an honor and also extremely useful to be able to interact with Google researchers through the meetings and summits in Zurich and Mountain View.

During my stay at LCL, I have been lucky to be able to collaborate with many other PhD students and researchers on a daily basis: Alessandro, Claudio, Daniele, Francesco, Ignacio, Massimiliano, Taher, Tiziano, Tommaso, Valentina... At this point I have to especially thank the help of Taher, who guided me from the beginning and taught me what research is about. I would also like to mention the folks at Pompeu Fabra University, with whom I have closely interacted and collaborated. In particular with Luis, who also visited us in Rome in 2015.

Finally, I have to deeply thank the support of friends and family, especially my parents who are always there for me, and of course my wife, who has accompanied me throughout the entire journey.

My PhD was financially supported by the ERC Starting Grant MultiJEDI No. 259234 (first year) and then by a Google PhD Fellowship. I am profoundly grateful to both for funding and supporting my research.

Contents

Chapter 1

Introduction

Effectively representing meaning is a key challenge in Natural Language Processing (NLP) and Artificial Intelligence (AI). In particular, the understanding of lexical items lies at the core of how humans process and generate language. Therefore, this is a crucial aspect in AI and NLP on their common overarching goal of enabling machines to understand language. Research on semantically representing lexical items dates back to the early days of NLP (Firth, 1957; Salton et al., 1975) and has agglutinated a large body of work since then. In fact, most current approaches are still based on the principles of Harris (1954), which called for an interpretation of the context in order to understand the meaning of words.

The most prominent research on this area of semantics nowadays is based on Vector Space Models (Turney and Pantel, 2010, VSM). In particular, research has in the main focused on representing words as points in a vector space. Recently, advances based on neural networks (i.e. word embeddings) have increased the popularity of this mainstream technique by successfully embedding words into low-dimensional vector spaces (Mikolov et al., 2013a; Pennington et al., 2014). Word embeddings have been successfully integrated into different neural architectures to provide a generalization boost on many applications such as machine translation (Zou et al., 2013), syntactic parsing (Weiss et al., 2015), and Question Answering (Bordes et al., 2014), to name a few.

However, word embeddings are hampered by an important limitation, as they conflate the different meanings of the same word into a single vector representation. In other words, they do not handle language lexical ambiguity. This is one of the main reasons why we propose to go beyond the word level by modeling senses instead of words. In fact, the waves of the word embedding tsunami have also lapped on the shores of sense representation, as several techniques for extending word embedding models to cluster contexts and induce senses from text corpora have been proposed (Huang et al., 2012; Neelakantan et al., 2014; Tian et al., 2014; Li and Jurafsky, 2015; Vu and Parker, 2016; Qiu et al., 2016). These techniques are usually referred to as unsupervised sense representations. While these unsupervised models are characterized by their adaptability to different domains, their induced senses are in the main hardly interpretable (Panchenko et al., 2017) and not easy to map to lexical resources, which limits their application.

In this thesis we explore the use of knowledge resources for an effective representation of lexical items going beyond the word level. Because they represent the lowest linguistic level, word senses play a vital role in natural language understanding. Effective representations of word senses have been proved to be directly useful in tasks such as Word Sense Disambiguation (Chen et al., 2014; Rothe and Schütze, 2015), semantic similarity (Budanitsky and Hirst, 2006; Turney and Pantel, 2010; Pilehvar and Navigli, 2015), coarsening of sense inventories (Snow et al., 2007; Pilehvar et al., 2013), alignment of lexical resources (Niemann and Gurevych, 2011; Navigli and Ponzetto, 2012; Pilehvar and Navigli, 2014a), lexical substitution (McCarthy and Navigli, 2009; Cocos et al., 2017a), and semantic priming (Neely et al., 1989). In particular, in this thesis we study the complementarities of both encyclopedic and lexicographic resources, putting forward various techniques for building flexible semantic representations for word senses, concepts and entities. This knowledge transferring has additional advantages that we also investigate in our work, with a marked emphasis in multilinguality. Acting as a bridge between the knowledge encoded in lexical resources and NLP applications, these knowledge-based representations pave the way for new lines of applied research.

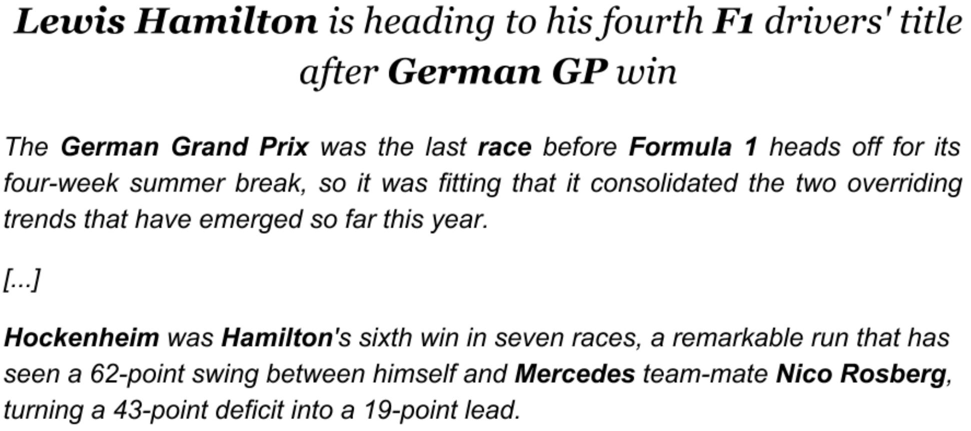
Finally, we show how these representations can be easily integrated into diverse applications, from word sense disambiguation to downstream NLP tasks such as text categorization and sentiment analysis. Additionally, we propose frameworks for leveraging these sense representations in applications aiming at improving the quality of current knowledge resources (sense clustering, domain labeling, hypernym discovery and collocation discovery), establishing an interesting interplay between lexical resources and NLP applications. For these applications we provide their background and position our methods with respect to the state of the art, which makes each of these sections self-sufficient as they provide interesting insights on their own. The versatility, flexibility and applicability of our proposed knowledge-based representations is largely proved throughout an extensive list of experiments.

1.1 Motivation

With the help of an example news article from the BBC, shown in Figure 1.1, we highlight some of the potential deficiencies of word-based NLP models which can be directly or indirectly solved by modeling senses:

Ambiguity. Language is inherently ambiguous. For instance, *Mercedes*, *race*, *Hamilton* and *Formula* can refer to different entities or meanings. Current neural models have managed to successfully represent complex semantic associations by effectively analyzing large amounts of data. However, the word-level functionality of these systems is still a barrier to achieve a deep natural language understanding. This is one of the main deficiencies of word-bases models. Since in our pipeline we model senses instead of words, our proposal is particularly tailored towards addressing this issue.

Multiword expressions. MultiWord Expressions (MWE) are lexical units made up of two or more words which are idiosyncratic in nature (Sag et al., 2002), e.g, *Lewis Hamilton*, *Nico Rosberg* and *Formula 1*. Most existing word-based models



***Lewis Hamilton** is heading to his fourth **F1** drivers' title after **German GP** win*

*The **German Grand Prix** was the last **race** before **Formula 1** heads off for its four-week summer break, so it was fitting that it consolidated the two overriding trends that have emerged so far this year.*

[...]

***Hockenheim** was **Hamilton**'s sixth win in seven races, a remarkable run that has seen a 62-point swing between himself and **Mercedes** team-mate **Nico Rosberg**, turning a 43-point deficit into a 19-point lead.*

Figure 1.1. Excerpt of a news article from the BBC.

ignore the interdependency between MWE's subunits and treat them as individual units. Handling MWE has been a long-standing problem in NLP and has recently received a considerable amount of interest (Tsvetkov and Wintner, 2014; Salehi et al., 2015). Our pipeline facilitates this goal thanks to the modeling of lexical items as represented in a given sense inventory, including instances consisting of multiple tokens.

Co-reference. Co-reference resolution of concepts and entities is not explicitly tackled by our approach. However, thanks to the fact that words that refer to the same meaning in context, e.g., *Formula 1-F1* or *German Grand Prix-German GP-Hockenheim*, are all disambiguated to the same concept, the co-reference issue is also partly addressed by explicitly modeling sense-level information.

Out-Of-Vocabulary words. Out-Of-Vocabulary (OOV) words are a recurring problem when using word-based models. As word vector representations are utilized in a wide variety of problems and domains, it is not rare to encounter these OOV words in practice (e.g. domain-specific words like *F1* or *Rosberg*). However, many OOV words are already represented in high-coverage lexical resources (see Section 2). Therefore, the use of these knowledge resources (e.g. WordNet, Wikipedia) for learning semantic representations could partially solve this problem. As we also argue throughout the thesis, a shared vector space of both words and senses would further contribute to expand this coverage, exploiting the best of both worlds.

1.2 Objectives

The main goal of this thesis is to advance the research in the lexical semantics field, and in general, on natural language processing. The concrete objectives of this thesis are listed below:

- Deal with language ambiguity through three inter-connected paths: (1) developing semantic representations at the sense, concept and entity levels; (2)

improving word sense disambiguation; and (3) enriching knowledge resources.

- Investigate the use of knowledge resources for the development of semantic representations and their integration into NLP applications.
- Explore the complementarities of using both encyclopedic and lexicographic resources.
- Investigate the potential of concept and entity representations in multilingual and cross-lingual environments.
- Develop benchmarks for evaluating word and sense vector representations intrinsically and extrinsically.
- Study the integration of sense representations into NLP applications, in particular word sense disambiguation, knowledge base enrichment and text classification.

1.3 Contributions and outline

The remainder of the thesis is organized as follows. We first present an overview of the main knowledge resources utilized in this thesis in **Chapter 2**.

Then, we move on to the **first part** of the thesis on semantic representation learning. In **Chapter 3** we present an overview of the related work on semantic representations, both at the word level and with a special focus on the works dealing with representations at the sense level. In the subsequent chapters we present two complementary models for learning semantic representations, Nasari and SW2V:

- In **Chapter 4** we present Nasari (a Novel Approach to a Semantically-Aware Representation of Items), which is one of the main contributions of this thesis. Nasari is based on the seamless integration of the encyclopedic and lexicographic resources presented in Chapter 2. We provide different types of semantic vector representation for concept and entities, going beyond the shallower word level. In addition to aiming at solving the ambiguity issue of word representations, Nasari provides three additional key features: (1) multilinguality, as the approach exploits at best the language diversity of Wikipedia and BabelNet, (2) high-coverage of both concepts and named entities as both WordNet and more importantly the full Wikipedia are covered, and (3) flexibility, as we put forward various vector representations which are easy to integrate into different applications as presented in the second part of the thesis.
- In **Chapter 5** we present SW2V (Senses and Words To Vectors), a flexible architecture for jointly learning word and sense embeddings on a single joint training phase. SW2V is complementary to Nasari as they learn from different signals. While Nasari provides representations for concept and entities from knowledge bases, SW2V learn word and sense embeddings directly from text

corpora. The main feature of SW2V lies on its flexibility and adaptability, as it can learn representations given any text corpora and semantic network.

The **second part** of the thesis focuses on the applications of these flexible semantic representations of senses, concepts and entities. We present the individual applications from Chapter 6 to 8:

- Word Sense Disambiguation (**Chapter 6**). Word Sense Disambiguation is a long-standing task in Artificial Intelligence and Natural Language Processing. In this chapter we present a knowledge-based approach for word and named entity disambiguation. The approach heavily relies on Nasari and, despite its simplicity, achieves results in line with the state of the art for several languages and resources, while showing interesting complementarities with supervised systems. Additionally, we show how Nasari can be leveraged for a high-confidence disambiguation of concepts and entities in multiple languages by leveraging comparable corpora. Thank to this approach we released two sense-annotated multilingual corpora: a large corpus of textual definitions and the Europarl parallel corpus.

- Knowledge-based Enrichment (**Chapter 7**). One of the advantages of exploiting knowledge-based sense vector representations is their ability to encode relevant semantic properties from lexical items in knowledge resources. This, coupled with their sense-based nature, enable them to accurately incorporate different sources of information stored on their vector representations. Being represented in a mathematical vector space, they can be effectively leveraged for improving and enriching knowledge resources. In particular, in this chapter we present four applications within this broad area: sense clustering, domain labeling, hypernym discovery and collocation discovery.

- Downstream NLP applications (**Chapter 8**). In this Chapter we present a simple approach to integrate senses into neural network architectures for downstream applications (Pilehvar et al., 2017). The integration is based on a pre-WSD step along with the use of pre-trained knowledge-based sense representations from WordNet and Wikipedia (Nasari). The evaluation is focused on two text classification tasks (i.e. topic categorization and sentiment analysis) and shows the potential of moving from the standard word-level to the sense-level on downstream applications. Our analysis also highlights the main advantages of this sense-based pipeline (and also its current weaknesses).

Finally, we present the conclusions and future work in **Chapter 9**.

1.4 Publications

A wide portion of the content of this thesis has already been published in relevant peer-reviewed conferences and journals from Natural Language Processing and Artificial Intelligence. The content of some of these publications represent the core of this thesis and their insights are included at great extent in some chapters or sections. We present the list of all the publications contributing to the overall thesis below:[1]

- Jose Camacho-Collados, Mohammad Taher Pilehvar and Roberto Navigli. 2016. *Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities.* Artificial Intelligence Journal, 240, pp. 36-64. **Chapter 4**.

- Massimiliano Mancini*, Jose Camacho-Collados*, Ignacio Iacobacci and Roberto Navigli. *Embedding Words and Senses Together via Joint Knowledge-Enhanced Training.* Proceedings of CoNLL, Vancouver, Canada, pp. 100-111. **Chapter 5**.

- Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli and Nigel Collier. 2017. Towards a Seamless Integration of Word Senses into Downstream NLP Applications. Proceedings of ACL, Vancouver, Canada, pp. 1857-1869. **Chapter 8**.

- Jose Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato and Roberto Navigli. 2017. *SenseDefs: A Multilingual Corpus of Sense-annotated Textual Definitions.* Language Resources and Evaluation Journal, *accepted.* **Section 6.3**.

- Jose Camacho-Collados and Roberto Navigli. 2017. *BabelDomains: Large-Scale Domain Labeling of Lexical Resources.* Proceedings of EACL (2), Valencia, Spain, pp. 223-228. **Section 7.2**.

- Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi and Horacio Saggion. 2016. *Supervised Distributional Hypernym Discovery via Domain Adaptation.* Proceedings of EMNLP, Austin, USA, pp. 424-435. **Section 7.3**.

- Luis Espinosa-Anke, Jose Camacho-Collados, Sara Rodríguez-Fernández, Horacio Saggion and Leo Wanner. 2016. *Extending WordNet with Fine-Grained Collocational Information via Supervised Distributional Learning.* Proceddings of COLING, Osaka, Japan, pp. 3422-3432. **Section 7.4**.

- Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato and Roberto Navigli. 2017. EuroSense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text. Proceedings of ACL (2), Vancouver, Canada, pp. 594-600. **Section 6.3.3**.

- Jose Camacho-Collados, Mohammad Taher Pilehvar and Roberto Navigli. 2015. *NASARI: a Novel Approach to a Semantically-Aware Representation of Items.* Proceedings of NAACL, Denver, USA, pp. 567-577.

[1] The publications with a higher contribution to the thesis are presented on top, and their corresponding chapters or sections are indicated accordingly. The rest of the publications, while having a more reduced space in the thesis, they have contributed to the overall dissertation and are acknowledged accordingly.

- Jose Camacho-Collados, Mohammad Taher Pilehvar and Roberto Navigli. 2015. *A Unified Multilingual Semantic Representation of Concepts.* Proceedings of ACL, Beijing, China, pp. 741-751.

- Jose Camacho-Collados, Mohammad Taher Pilehvar and Roberto Navigli. 2015. *A Framework for the Construction of Monolingual and Cross-lingual Word Similarity Datasets.* Proceedings of ACL (2), Beijing, China, pp. 1-7.

- Jose Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato and Roberto Navigli. 2016. *A Large-Scale Multilingual Disambiguation of Glosses.* Proceedings of LREC, Portoroz, Slovenia, pp. 1701-1708.

- Jose Camacho-Collados and Roberto Navigli. 2016. *Find the word that does not belong: A Framework for an Intrinsic Evaluation of Word Vector Representations.* Proceedings of the ACL Workshop on Evaluating Vector Space Representations for NLP, Berlin, Germany, pp. 43-50.

- Alessandro Raganato, Jose Camacho-Collados and Roberto Navigli. 2017. *Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison.* Proceedings of EACL, Valencia, Spain, pp. 99-110.

- Jose Camacho-Collados*, Mohammad Taher Pilehvar*, Nigel Collier and Roberto Navigli. 2017. *SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity.* Proceedings of the International Workshop on Semantic Evaluation (SemEval), Vancouver, Canada, pp. 15-26.

Chapter 2

Preliminaries: Knowledge Resources

In this thesis we investigate how to integrate knowledge from lexical resources for representing senses, concepts and entities and use them as a bridge on NLP applications. In this chapter we describe the main knowledge resources utilized in our work.

Knowledge resources can be split into two main groups: expert made and collaboratively constructed. Expert or manually constructed resources feature highly-accurate encoding of concepts and semantic relationships between them but, with a few exceptions, are usually limited in their lexical coverage, and are typically focused on a specific language only. On the other hand, collaboratively-constructed resources encode a different source of information, e.g. extracted from large amounts of textual data. For this reason they tend to be less accurate but provide a different set of features such as a higher coverage and multilinguality. Likewise, knowledge resources can encode different kind of knowledge, i.e. lexicographic or encyclopedic. Each type has its own advantages and limitations. Some of these resources, i.e. WordNet (Section 2.1) or BabelNet (Section 2.3) which are used in this work, are structured as graphs, with nodes denoting concepts and edges representing relations between concepts. Resources containing a diverse set of relations are known as *knowledge bases*.

In our work we attempt to take the best of all these types of resource by combining their most prominent features. In particular, we make use of an expert-made lexicographic resource, i.e. WordNet (Section 2.1), and a collaboratively-constructed encyclopedic resource, i.e. Wikipedia (Section 2.2). In order to combine the knowledge from these two resources, we make use of BabelNet (Section 2.3) as a bridge. We additionally present other lexical resources used in this thesis in Section 2.4 and some common notions and nomenclature from these knowledge resources in Section 2.5.

2.1 WordNet

A prominent example of expert-made resource is **WordNet** (Miller et al., 1990; Miller, 1995), a semantic network whose basic units are synsets. A synset represents a concept which may be expressed through nouns, verbs, adjectives or adverbs and is composed of one or more lexicalizations (i.e., synonyms that are used to express it). For example, the synset of the *middle of the day* concept consists of six lexicalizations: *noon, twelve noon, high noon, midday, noonday, noontide.* Synsets may also be seen as nodes in a semantic network. These nodes are connected to each other by means of lexical or semantic relations (hypernymy, meronymy, etc.). These relations are seen as the edges in the WordNet semantic network. Despite being one of the largest and most complete manually-made lexical resources, WordNet still lacks coverage of lemmas and senses from domain specific lexicons (e.g., law or medicine), named entities, slang usages, or those for technology that came into existence only recently. In our experiments we used WordNet 3.0, which covers more than 117K unique nouns in about 80K synsets.

2.2 Wikipedia

Wikipedia is one of the the most prominent examples of collaboratively-constructed resources. It provides features such as multilinguality (Wikipedia covers more than 250 languages), wide coverage, encyclopedic information and up-to-dateness. As of September 2015, Wikipedia provided more than 100K articles in over fifty languages. This coverage is steadily increasing. For instance, the English Wikipedia alone receives 750 new articles per day. Each of these articles provides, for its corresponding concept, a great deal of information in the form of textual information, tables, infoboxes, and various relations (such as redirections, disambiguations, and categories). These features have persuaded many researchers over the past years to exploit the large amounts of semi-structured knowledge available in such collaborative resources for different NLP applications (Cucerzan, 2007; Gabrilovich and Markovitch, 2007; Wu and Weld, 2010; Hovy et al., 2013; Wu and Giles, 2015; Søgaard et al., 2015). In this work we utilized the Wikipedia dump from November 2014 in multiple languages.

2.3 BabelNet

The types of knowledge available in the expert-based and collaboratively-constructed resources make them complementary. This has motivated researchers to combine various lexical resources across the two categories (Niemann and Gurevych, 2011; Pilehvar and Navigli, 2014a). A prominent example is **BabelNet** (Navigli and Ponzetto, 2012), which provides a mapping of WordNet to a number of collaboratively-constructed resources, including Wikipedia. The structure of BabelNet[1] is similar to that of WordNet. Synsets are the main linguistic units and are connected to other semantically related synsets, whose lexicalizations are multilingual in this case. For instance, the synset corresponding to *United States* is represented with a set of multilingual lexicalizations including $United\_States_{EN}$, $United\_States\_of\_America_{EN}$, $America_{EN}$, $U.S._{EN}$, *and* $U.S.A._{EN}$ in English, $Estados\_Unidos_{ES}$, $Estados\_Unidos\_de\_América_{ES}$, $EEUU_{ES}$, $E.E.U.U._{ES}$, *and*

[1] http://babelnet.org/

EE. UU.$_{ES}$ in Spanish, and *Stati_Uniti_d'America*$_{IT}$*, Stati_Uniti*$_{IT}$*, America*$_{IT}$*, and U.S.A.*$_{IT}$ in Italian. The relations between synsets are come from WordNet (hypernyms, hyponyms, etc.), plus new semantic relations coming from other resources such as Wikipedia hyperlinks and Wikidata relations (e.g. Madrid *capital of* Spain). BabelNet is the largest multilingual semantic network available, containing 13,789,332 synsets (6,418,418 concepts and 7,370,914 named entities) and 354,538,633 relations for 271 languages[2]. For the English language, BabelNet contains 4,403,148 synsets with at least one Wikipedia page associated and 117,653 synsets with one WordNet synset associated, from which 99,705 synsets are composed of both a Wikipedia page and a WordNet synset.

2.4 Other resources

Wikidata[3] (Vrandečić and Krötzsch, 2014) is a document-oriented semantic database operated by the Wikimedia Foundation with the goal of providing a common source of data that can be used by other Wikimedia projects (e.g. Wikipedia). It is designed as a document-oriented semantic database based on *items*, each representing a concept or an entity and associated with a unique identifier. Knowledge is encoded with *statements* in the form of property-value pairs, among which definitions (*descriptions*) are also included. Wikidata is also integrated into BabelNet, containing over 8 million definitions[4]. In this thesis we use Wikidata both for obtaining term-hypernym pairs (Section 7.3) and as a repository of definitions (Section 6.3.2).

Beyond these resources, we also use two collaborative multilingual dictionaries (i.e. Wiktionary and OmegaWiki) as an additional repository of definitions. **Wiktionary**[5] is a Wikimedia project designed to represent lexicographic knowledge that would not be well suited for an encyclopedia (e.g. verbal and adverbial senses). It is available for over 500 languages typically with a very high coverage, including domain-specific terms and descriptions that are not found in WordNet. Similar to Wiktionary, **OmegaWiki**[6] is a large multilingual dictionary based on a relational database, designed with the aim of unifying the various language-specific Wiktionaries into a unified lexical repository.

2.5 Nomenclature

In this thesis we mainly follow the same nomenclature of WordNet and BabelNet. In these resources a ***synset*** refer to an specific meaning as defined on the knowledge resource (e.g. the concept of *middle of the day* mentioned above) which may have more than one or more words as its associated lexicalizations (e.g. *noon, twelve noon, high noon, midday, noonday, noontide* for the concept of *middle of the day*). Synsets may be ***concepts*** (e.g. *middle of the day*) or ***entities*** (e.g. *Microsoft*). In

[2] The statistics are taken from the BabelNet 3.0 release, which is the version used in our experiments. More statistics can be found at `http://babelnet.org/stats`

[3] `https://www.wikidata.org`

[4] Given its strictly computational nature, it often provides minimal definition phrases containing only the superclass of the definiendum.

[5] `https://www.wiktionary.org`

[6] `http://www.omegawiki.org`

general, when we refer to a synset and if it is not explicitly mentioned, we refer to both concepts and entities. ***Sense*** may refer to the general term including all representations going beyond the word level, or explicitly to a word associated with a specific meaning (e.g. *noon* with its meaning *middle of the day*)[7], irrespective of whether the meaning belongs to a pre-defined sense inventory or not. Following previous works (Navigli, 2009), we use the following notation for senses: $word_n^p$ is the n^{th} sense of *word* with part of speech p.

[7] In other works *senses* have also been referred to as *lexemes* (Rothe and Schütze, 2015).

PART 1: Semantic Representation of Senses, Concepts and Entities

In this initial part of the thesis we focus on the semantic representation of word senses, concepts and entities. We investigate and propose theoretical frameworks for learning these fine-grained semantic representations. In particular, we concentrate on the research area making use of knowledge resources for enhancing representation learning. This first block is structured as follows.

First, we provide a comprehensive overview on previous work on word and sense representations (Chapter 3). We begin this background chapter by presenting and describing some fundamental theories of word-level representation learning. Then, inspired by these grounding works, we present more recent related works on the semantic representation of lexical items of a finer granularity, i.e., senses.

Following this background chapter we move on to explain our work on sense representation learning. We present two theoretical models for learning sense representations: Nasari (Chapter 4), a purely knowledge-based method which exploits the complementarities of various knowledge resources and text corpora; and SW2V (Chapter 5), a method for jointly training word and sense embeddings from text corpora exploiting semantic networks. These two methods, exploiting distinct signals, learn different kinds of semantic representation, providing advantages and disadvantages depending on their application[8].

[8] In the second part of this thesis we will present a set of relevant applications for which these semantic representations prove their suitability.

Chapter 3

From Word to Sense Representations: Background

In this chapter we present an overview of related approaches on semantic representation learning. In Section 3.1 we describe mainstream techniques for word representation learning, while in Section 3.2 we briefly review the recent literature on the representation of word senses, which is the main topic of this thesis.[1]

3.1 Word Representations

Word representation learning constitutes one of the main research topics in semantics since the origin of NLP. The most prominent methods are in the main based on the Vector Space Model (VSM), which is in turn supported by research in human cognition (Landauer and Dumais, 1997; Gärdenfors, 2004). The earliest use of this model considered a document as a vector whose dimensions were the vocabulary (Salton et al., 1975). Weights of individual dimensions were initially based on its corresponding word's frequencies within the document. Different weights have also been explored, but mainly based on frequencies or normalized frequencies (Salton and McGill, 1983). This methodology has been successfully refined and applied in many NLP applications such as information retrieval (Lee et al., 1997) text classification (Soucy and Mineau, 2005), or sentiment analysis (Turney, 2002), to name a few. Turney and Pantel (2010) provides a comprehensive overview of VSM and their applications.

This document-based VSM has also been extended to other lexical items like words. On this case a word is represented as a point in the vector space. A word-based vector is traditionally constructed based on the normalized frequencies of the co-occurring words within a corpus (Lund and Burgess, 1996), trailing the initial theories of Harris (1954). These models have also proved effective in NLP tasks such as information extraction (Laender et al., 2002), semantic role labeling (Erk, 2007), word similarity (Radinsky et al., 2011), WSD (Navigli, 2009) or spelling correction (Jones and Martin, 1997), *inter alia*.

[1] For a complementary overview on this topic we recommend our ACL 2016 tutorial on "Semantic Representations of Word Senses and Concepts" (Camacho-Collados et al., 2016b).

One of the main drawbacks of these approaches is the high dimensionality of the produced vectors. Since the dimensions correspond to words in the vocabulary, this number could easily add to the hundreds of thousands or even millions, depending on the underlying corpus. The most recurrent approaches for dimensionality reduction make use of the Singular Value Decomposition (SVD) and are known as Latent Semantic Analysis (Hofmann, 2001; Landauer and Dooley, 2002, LSA)

A more recent trend exploit neural network architectures to embed words into low-dimensional vectors. These models are commonly known as *word embeddings*. The earliest attempts of this kind were built upon the probabilistic feedforward neural network language model Bengio et al. (2003). This approach was popularized through Word2Vec (Mikolov et al., 2013a), proposing a simplified architecture and showed some interesting semantic properties of the output vectors (Mikolov et al., 2013d). Another prominent word embedding architecture is GloVe (Pennington et al., 2014), combining global matrix factorization and local context window methods through a bilinear regression model. Word embeddings have been shown to provide a valuable prior knowledge which have been proved decisive for achieving state-of-the-art performance in many NLP tasks when integrated into a neural network architecture (Zou et al., 2013; Kim, 2014; Bordes et al., 2014; Weiss et al., 2015).

However, word representations fail to capture the meaning of the various senses of the same word (Yaghoobzadeh and Schütze, 2016). In the following section we present approaches attempting to overcome this limitation by modeling senses instead of words.

3.2 Sense Representations

While most research studies in semantic representation have so far concentrated on the representation of words, few studies have focused on the representation of word senses or concepts. This is partly due to the so-called knowledge acquisition bottleneck that arises because the application of distributional word modeling techniques at the sense level would require the availability of high-coverage sense-annotated data. However, word representations are known to suffer from some issues which dampen their suitability for tasks that require accurate representations of meaning. The most important drawback with word representations lies in their inability to model polysemy and homonymy, as they conflate different meanings that a word can have into a single representation (Tversky and Gati, 1982; Reisinger and Mooney, 2010). For instance, a word representation for the word *bank* does not distinguish between its financial institution and river bank meanings (the noun *bank* has ten senses according to WordNet 3.0). Approaches which leverage semantic lexicons to improve word representations (Yu and Dredze, 2014; Faruqui et al., 2015; Goikoetxea et al., 2015; Speer and Lowry-Duda, 2017; Mrkšic et al., 2017) also suffer from the same drawback, as their target modeling units are still potentially-ambiguous words.

Because they represent the lowest linguistic level, word senses and concepts play a crucial role in natural language understanding. Meanings of a word are identified and separately modeled, the resulting representations are ideal for performing an accurate semantic processing. In addition, the fine-grained representation of word senses can be directly extended to higher linguistic levels (Budanitsky and Hirst, 2006), which makes them particularly interesting. These features have recently attracted the attention of different research studies. Most of these techniques view sense representation as a specific type of word representation and try to adapt the existing distributional word modeling techniques to the sense level, usually through clustering the contexts in which a word appears (Schütze, 1998; Reisinger and Mooney, 2010; Huang et al., 2012). The main fundamental assumption of these works is that the intended meaning of a word mainly depends on its context and hence one can obtain sense-specific contexts for a given word sense by clustering the contexts in which the word appears in a given text corpus. Various clustering-based techniques usually differ in their clustering procedure and how this is combined with the representation technique. For instance, while some approaches rely on monolingual corpora only (Neelakantan et al., 2014; Tian et al., 2014; Li and Jurafsky, 2015; Vu and Parker, 2016; Qiu et al., 2016), other works rely on bilingual or multilingual corpora for building their sense vector representations (Guo et al., 2014; Ettinger et al., 2016; Šuster et al., 2016; Upadhyay et al., 2017). However, all these models are often limited to representing only those senses that are covered in the underlying corpus. Moreover, the sense representations obtained using these methods are generally not interpretable (Panchenko et al., 2017) and usually not linked to any sense inventory. Therefore such linking has to be carried out, either manually, or with the help of sense-annotated data if the representations are to be used for direct applications such as Word Sense Disambiguation.

Most sense modeling techniques have based their representation on the knowledge derived from resources such as WordNet. Earlier techniques exploit the information provided in WordNet, such as the synonymous words in a synset, for the representation of word senses (Mihalcea and Moldovan, 1999; Agirre and de Lacalle, 2004). More recent approaches usually adapt distributional models to the sense level on the basis of lexico-semantic knowledge derived from lexical resources such as Wikipedia (Gabrilovich and Markovitch, 2007; Mihalcea, 2007), WordNet (Chen et al., 2014; Jauhar et al., 2015; Rothe and Schütze, 2015) or other language-specific semantic networks (Johansson and Pina, 2015). WordNet can be also viewed as a semantic network where its individual synsets are represented on the basis of graph-based algorithms (Pilehvar et al., 2013; Pilehvar and Collier, 2016). Word Sense Disambiguation of large amounts of textual data has also been explored as a means of obtaining high-coverage annotated data for learning sense representations based on neural networks (Iacobacci et al., 2015), representations referred to as sense embeddings. Chen et al. (2014), which uses WordNet as main knowledge source, also relies on WSD for obtaining their sense representations. However, these two approaches are hampered by their inherently imperfect WSD systems.

Additionally, these techniques are often limited to the reduced coverage of WordNet and to the English language only. In contrast, our proposed methods (see Chapter 4) provide a multilingual representation of word senses on the basis of the complementary knowledge of two different resources, enabling a significantly higher coverage of specific domains and named entities. In addition, the synset representations proposed are inherently multilingual, which open up new possibilities for their application in multilingual and cross-lingual applications.

Moreover, the representation of words and senses in the same vector space has been proved key for applying these knowledge-based sense vector representations in downstream applications, particularly for their integration into neural architectures (Pilehvar et al., 2017) via word and sense embeddings. In the literature, various different methods have attempted to build a shared space of word and sense vectors. Chen et al. (2014) proposed a model for obtaining both word and sense embeddings based on a first training step of conventional word embeddings, a second disambiguation step based on sense definitions, and a final training phase which uses the disambiguated text as input. Likewise, Rothe and Schütze (2015) aimed at building a shared space of word and sense embeddings based on two steps: a first training step of only word embeddings and a second training step to produce sense and synset embeddings. These two approaches require multiple steps of training and make use of a relatively small resource like WordNet, which limits their coverage and applicability. Wang et al. (2014) and Fang et al. (2016) increased the coverage of these WordNet-based approaches by exploiting larger resources like Wikipedia, proposing a model to align vector spaces of words and entities from knowledge bases. However, these approaches are restricted to nominal instances only (i.e. Wikipedia pages or entities). In contrast, we propose a model (see Chapter 5) which learns both words and sense embeddings from a single joint training phase, producing a common vector space of words and senses as an emerging feature.

Chapter 4

NASARI: Multilingual Representations of Concepts and Entities

In this chapter we describe the methodology for constructing semantic representations of concepts and entities. Our approach, referred to as NASARI (Novel Approach to a Semantically-Aware Representation of Items), exploits the structural knowledge derived from semantic networks, along with distributional statistics from text corpora, to produce effective representations of individual word senses or concepts. Our method provides two main advantages in comparison to previous VSM techniques: (1) it is multilingual, as it can be directly applied for the representation of concepts in dozens of languages; and (2) each vector represents a concept, irrespective of its language, in a unified semantic space having words or concepts as its dimensions, permitting direct comparison of different representations across languages and hence enabling cross-lingual applications.

The rest of this chapter is structured as follows. First, in Section 4.1, we describe our methodology to convert text into lexical, embedded and unified vectors. The process to obtain vector representations for synset vectors by leveraging the knowledge resources described in Chapter 2 is presented in Section 4.2. In Section 4.3 we perform an intrinsic evaluation of these vectors. We analyze the performance of different components of our model in Section 4.4. Finally, we provide the concluding remarks in Section 4.5.

4.1 Representing texts as vectors

One of the contributions of this chapter is the framework we are proposing for transforming texts into three different kinds of vector: lexical, embedded and unified. Our lexical vectors follow the conventional approach for representing a linguistic item in a semantic space with words[1] as its dimensions (Pantel and Lin, 2002). The weights in these vectors are usually computed on the basis of raw term frequencies (*tf*) or normalized frequencies, such as *tf-idf* Jones (1972). Instead, we use lexical

[1] Short noun phrases and multiword expressions are also considered in our work.

specificity for the computation of the weights in our lexical vectors. Having a solid statistical basis, lexical specificity provides several advantages over the previously mentioned measures (see Section 4.4 for a comparison between lexical specificity and *tf-idf*). In what follows in this section we first explain lexical specificity and propose an efficient way for its fast computation (Section 4.1.1). We then provide more details of our three types of vector, i.e., lexical (Section 4.1.2), embedded (Section 4.1.3) and unified (Section 4.1.4).

4.1.1 Lexical specificity

Lexical specificity (Lafon, 1980) is a statistical measure based on the hypergeometric distribution[2]. The measure has been widely used in different NLP applications including term extraction (Drouin, 2003), textual data analysis (Lebart et al., 1998) and domain-based term disambiguation (Camacho-Collados et al., 2014; Billami et al., 2014), but it has rarely been used to measure weights in a vector space model. Lexical specificity essentially computes the set of most representative words for a given text based on the hypergeometric distribution. In our setting, we are interested in representing a given text, hereafter referred to as the sub-corpus $\mathcal{SC}$, through a vector comprising the weighted set of its most relevant words or concepts. In order to compute lexical specificity, we need a reference corpus $\mathcal{RC}$ which should be a superset of $\mathcal{SC}$. Lexical specificity computes the weights for each word by contrasting the frequencies of that word across $\mathcal{SC}$ and $\mathcal{RC}$.

Following the notation of Camacho-Collados et al. (2015c), let T and t be the respective total number of content words in $\mathcal{RC}$ and $\mathcal{SC}$, while F and f denote the frequency of a given word w in $\mathcal{RC}$ and $\mathcal{SC}$, respectively. Our goal is to compute a weight quantifying the association strength of w with our text $\mathcal{SC}$. We compute the probability of a word w having a frequency equal to or higher than f in our sub-corpus $\mathcal{SC}$ using a hypergeometric distribution which takes as its parameters the frequency of w in the reference corpus $\mathcal{RC}$, i.e., F, and the sizes of $\mathcal{RC}$ and $\mathcal{SC}$, i.e., T and t, respectively. A word w with a high probability is one with a high occurrence chance across arbitrary subsets of $\mathcal{RC}$ of size t. Hence, the representative words of a given sub-corpus will be those with low probabilities since these specific words are the most suitable ones for distinguishing the sub-corpus from the reference corpus. As a result, the computed probability is inversely proportional to the relevance of the word w to $\mathcal{SC}$. In order to make the relation directly proportional, thus making the weights more interpretable, we apply the $-\log_{10}$ operation to the computed probabilities as has been customary in the literature (Drouin, 2003; Heiden et al., 2010). This logarithmic operation also speeds up the calculations (more details in the following section). Morover, using $\log_{10}$, instead of for instance the natural logarithm, has the added benefit of leading to an easy calculation of the prior probability. For example, if an item has a lexical specificity of 5.0, it means that the probability of

[2] "The hypergeometric distribution is a discrete probability distribution that describes the probability of k successes in n draws, without replacement, from a finite population of size N that contains exactly K successes, wherein each draw is either a success or a failure. In statistics, the hypergeometric test uses the hypergeometric distribution to calculate the statistical significance of having drawn a specific k successes (out of n total draws) from the aforementioned population" (`https://en.wikipedia.org/wiki/Hypergeometric_distribution`).

observing that item in $\mathcal{SC}$ is $10^{-5} = 0.00005$. Therefore, the lexical specificity of w in $\mathcal{SC}$ is given by the following expression:

$$spec(T, t, F, f) = -\log_{10} P(X \geq f) \tag{4.1}$$

where X represents a random variable following a hypergeometric distribution with parameters F, t and T and $P(X \geq f)$ is defined as follows:

$$P(X \geq f) = \sum_{i=f}^{F} P(X = i) \tag{4.2}$$

where $P(X = i)$ represents the probability of a given word to appear exactly i times in the subcorpus $\mathcal{SC}$ according to the hypergeometric distribution of parameters F, t and T. In the following we propose an efficient implementation of Equation 4.2.

Efficient implementation of lexical specificity

According to Equation 4.2, the computation of the hypergeometric distribution involves summing $(F - f) + 1$ addends, each of which is calculated as follows[3]:

$$P(X = i) = \frac{\binom{F}{i}\binom{T-F}{t-i}}{\binom{T}{t}} = \frac{F!(T-F)!t!(T-t)!}{T!i!(F-i)!(t-i)!(T-F-t+i)!} \tag{4.3}$$

Given that the summation range of Equation 4.2 is generally directly proportional to the size of the corpus, the computation of lexical specificity can be quite expensive on large corpora wherein the value of F tends to be very high. Lafon (1980) proposed a method to reduce the computation cost of Equation 4.2. According to this method, one can first calculate $P(X = i)$ only for the smallest i (i.e., f) and then calculate the rest of probabilities, i.e., $P(X = f + 1)$, ..., $P(X = F)$, using the following property of the hypergeometric distribution:

$$P(X = i + 1) = \frac{P(X = i)(F - i)(t - i)}{(i + 1)(T - F - t + i + 1)} \tag{4.4}$$

Lafon (1980) also suggested using the well-known Stirling formula for the computation of the factorial components in Equation 4.3. According to the Stirling formula, the logarithm of a factorial can be approximated as follows:

$$\log n! = n \log n - n + \frac{1}{2} \log(2\pi n) \tag{4.5}$$

Thanks to the application of the Stirling formula we can transform Equation 4.3 into a summation. Despite these improvements in the calculation of lexical specificity, there remain issues when the above computation is to be applied to a large reference corpus. One of the main problems is the multiplication of potentially very small quantities. Specifically, a 64-bit binary floating-point number, which is the one typically used in current computers, has an approximate range from 10^{-308} through 10^{+308}. During the computation of lexical specificity on large corpora, the lower bound can be reached several times. Our solution to solve this problem (which

[3] In the cases where $i > t$, which may occur if $F > t$, the probability $P(X = i)$ is equal to 0.

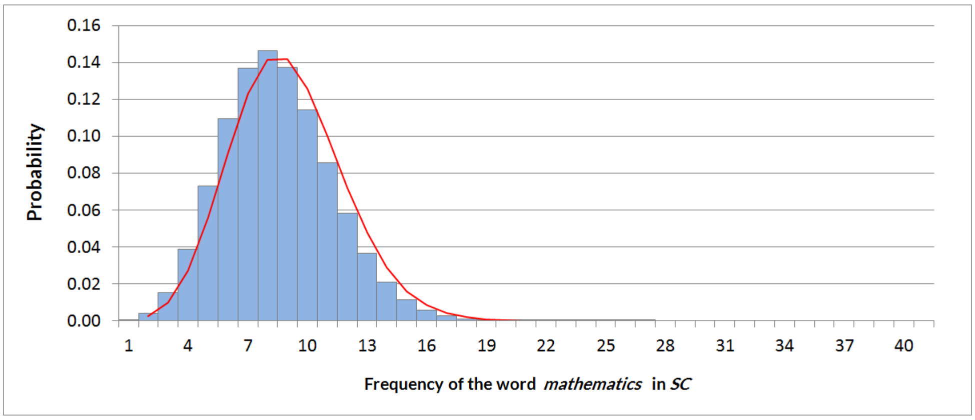


Figure 4.1. Hypergeometric distribution for the word *mathematics* in an arbitraty sub-corpus ($\mathcal{SC}$) of size 100,000 in Wikipedia.

even optimizes the calculations) is obtained via the next two equations. Firstly, we rewrite Equation 4.4 by extracting the common factor $P(X = f)$:

$$P(X \geq f) = \sum_{i=f}^{F} P(X = i) = P(X = f) \sum_{i=f}^{F} a_i \tag{4.6}$$

where $a_f = 1$ and $a_i = a_{i-1} \frac{(F-i)(t-i)}{(i+1)(T-F-t+i+1)}, \forall i = f+1, ..., F$.

Now we only need to apply the logarithm to both sides of the equation in order to transform the previous multiplication into an addition and thus avoid small values. In this way we also avoid unnecessary exponentials in the calculations of $P(X = f)$:

$$-\log_{10} P(X \geq f) = -\log_{10} P(X = f) - \log_{10} \left(\sum_{i=f}^{F} a_i \right) \tag{4.7}$$

Therefore, according to Equation 4.1 and by applying a change of logarithm base, we can compute lexical specificity given the four parameters T, t, F, and f as follows:

$$\mathit{spec}(T, t, F, f) = -k \log_e P(X = f) - \log_{10} \left(\sum_{i=f}^{F} a_i \right) \tag{4.8}$$

where k is the natural logarithm of 10 (i.e., $\log_e 10$).

For computational feasibility, the $\sum_{i=f}^{F} a_i$ sum is usually not computed until F. Instead, a stopping criterion is introduced into the loop. Since the probability mass in the tail of the hypergeometric distribution is in most cases mathematically insignificant with respect the final cumulative probability distribution, the stopping criterion is usually satisfied well before reaching to the final F value, which considerably reduces the computation time.

As an example we show in Figure 4.1 the estimated probability distribution for the word *mathematics* in an arbitrary sub-corpus $\mathcal{SC}$ of 100,000 content words from Wikipedia. If the word *mathematics* occurs more than twenty times in $\mathcal{SC}$, the word is considered to be very specific to the given subcorpus, since, as we can see from Figure 4.1, most of the probability mass in the hypergeometric distribution is concentrated in the left part of the distribution range. The distribution range extends until 70,029, which is the number of occurrences of the word *mathematics* in the whole Wikipedia. However, the probability $P(X = 45)$ is already as small as 10^{-20} and rapidly gets much smaller. This illustrates the point made above, in which the right tail of the probability mass is generally insignificant to values close to the expected value, and adding a stopping condition might make the calculations much faster, while not having any noticeable effect to the final specificity score.

The next three sections provide more details on our three types of vector and on how we leverage lexical specificity for their construction.

4.1.2 Lexical vector representation

So far we have explained how lexical specificity can be used to determine the relevance of words for a given text. In this section we explain how we leverage lexical specificity in order to construct a lexical vector for a given text (i.e., $\mathcal{SC}$). In this chapter all the texts considered come from Wikipedia, thus we use the whole Wikipedia as our reference corpus ($\mathcal{RC}$). Our lexical vectors have individual words as their dimensions, therefore, in our lexical semantic space, a text is represented on the basis of its association with a set of lexical items, i.e., words. By contrasting the term frequencies across $\mathcal{SC}$ and $\mathcal{RC}$, we compute the lexical specificity of each term for the given subcorpus.

Specifically, in order to compute our lexical vector $\vec{v}_{lex}(\mathcal{SC})$, we simply iterate over all the content words in our subcorpus $\mathcal{SC}$ (only words with a total frequency greater than or equal to five in the whole Wikipedia are considered) and compute lexical specificity for each of them. We then prune the resulting vectors by keeping only those words that are relevant to the target text with a confidence of 99% or more according to the hypergeometric distribution ($P(X \geq f) \leq 0.01$), as also performed in earlier works (Billami et al., 2014; Camacho-Collados et al., 2015b). Words with weights below the aforementioned threshold are considered as zero dimensions. The vector truncation step helps reduce noise. Additionally, the truncation helps in speeding up the computation of the vectors, as they will be sparse and therefore computationally easier to work with.

In our setting we also consider multiword expressions when they appear as lexicalizations of piped links[4]. Note that we apply lexical specificity to content words (nouns, verbs and adjectives) after tokenization and lemmatization, but for notational simplicity we will keep using the term "word" to refer to them.

4.1.3 Embedded vector representation

In recent years, semantic representation has experienced a resurgence of interest in the use of neural network-based learning, a trend usually referred to as word embeddings.

[4] A piped link is a hyperlink which is found within the Wikipedia article that redirects the user to another Wikipedia page. For example, the piped link [[dockside_crane|Crane_(machine)]] is a hyperlink that appears as *dockside_crane* in the text, but links to the Wikipedia page titled *Crane_(machine).* The Wikipedia article is therefore represented with a suitable lexicalization that preserves the grammatical and syntactic structure, the contextual coherency and the flow of the sentence.

In addition to their fast processing of massive amounts of text, word embeddings have proved to be reliable techniques for modeling the semantics of words on the basis of their contexts. However, the application of these word-based techniques to the representation of word senses is not trivial and is bound to the availability of large amounts of sense-annotated data. There have been efforts aimed at learning sense-specific embeddings without needing to resort to sense-annotated data, often through clustering the contexts in which a word appears (Weston et al., 2013; Huang et al., 2012; Neelakantan et al., 2014). However, the resulting representations are usually not aligned to existing sense inventories.

We put forward an approach that allows us to plug in an arbitrary word embedding representation with that of our lexical vector representations, providing three main advantages: (1) benefiting from the word-based knowledge derived as a result of learning from massive corpora for our sense-level representation; (2) reducing the dimensionality of our lexical space to a fixed-size continuous space; and (3) providing a shared semantic space between words and synsets (more details in Section 4.2), hence enabling a direct comparison of words and synsets.

Our approach exploits the compositionality of word embeddings. According to this property, a compositional phrase representation can be obtained by combining, usually averaging, its constituents' representations (Mikolov et al., 2013c). For instance, the vector representation obtained by averaging the vectors of the words *Vietnam* and *capital* is very close to the vector representation of the word *Hanoi* in the semantic space of word embeddings. Our approach builds on this property and plugs a trained word embedding-based representation into our lexical vectors.

Specifically, given an input text $\mathcal{T}$ and a space of word embeddings E, we first calculate the lexical vector of $\mathcal{T}$ (i.e., $\vec{v}_{lex}(\mathcal{T})$) as explained in Section 4.1.2 and then map our lexical vector to the semantic space E as follows:

$$E(\mathcal{T}) = \frac{\sum_{w \in \vec{v}_{lex}(\mathcal{T})} \left(\frac{1}{rank(w, \vec{v}_{lex}(\mathcal{T}))} E(w) \right)}{\sum_{w \in \vec{v}_{lex}(\mathcal{T})} \frac{1}{rank(w, \vec{v}_{lex}(\mathcal{T}))}} \tag{4.9}$$

where $E(w)$ is the embedding-based representation of the word w in E, and $rank(w, \vec{v}_{lex}(\mathcal{T}))$ is the rank of the dimension corresponding to the word w in the lexical vector $\vec{v}_{lex}(\mathcal{T})$, thus giving more importance to the higher weighted dimensions. In Section 4.4 we compare this harmonic average giving more importance to higher weighted words over a simple average. One of the main advantages of this representation combination technique is its flexibility, since any word embedding space can be given as input. As we show in our experiments in Section 4.3, this combination enables us to benefit from word-specific knowledge and improve it by integrating it into our sense-specific representations.

4.1.4 Unified vector representation

We additionally propose a third representation, which we call unified, that, in contrast to the lexical vector representation which has potentially ambiguous words as individual dimensions, has unambiguous BabelNet synsets as its individual dimensions. Algorithm 1 shows the construction process of a unified vector given the sub-corpus $\mathcal{SC}$. The algorithm first clusters together those words in $\mathcal{SC}$ that have a

Algorithm 1 Unified Vector Construction

```
Input: A reference corpus $\mathcal{RC}$ and a sub-corpus $\mathcal{SC}$
Output: the unified vector $\vec{u}_s$ where $\vec{u}_s(h)$ is the dimension corresponding to the synset $h$
 1: $T \leftarrow size(\mathcal{RC})$
 2: $t \leftarrow size(\mathcal{SC})$
 3: $H \leftarrow \emptyset$
 4: for each lemma $l \in \mathcal{SC}$
 5:     for each hypernym $h$ of $l$ in BabelNet
 6:         $H \leftarrow H \cup \{h\}$
 7: $\vec{u} \leftarrow$ empty vector
 8: for each $h \in H$
 9:     if $\exists\, l_1, l_2 \in \mathcal{SC}$: $l_1$, $l_2$ hyponyms of $h$ and $l_1 \neq l_2$ then
10:         $F \leftarrow 0$
11:         $f \leftarrow 0$
12:         $hyper_{pass} \leftarrow False$
13:         for each lexicalization $lex$ of $h$
14:             $F \leftarrow F + freq(lex, \mathcal{RC})$
15:             $f \leftarrow f + freq(lex, \mathcal{SC})$
16:             $spec_h \leftarrow specificity(T, t, freq(lex, \mathcal{RC}), freq(lex, \mathcal{SC}))$
17:             if $spec_h \geq spec_{thres}$ then
18:                 $hyper_{pass} \leftarrow True$
19:         if $hyper_{pass}$ then
20:             for each hyponym $hypo$ of $h$
21:                 for each lexicalization $lex$ of $hypo$
22:                     $F \leftarrow F + freq(lex, \mathcal{RC})$
23:                     $f \leftarrow f + freq(lex, \mathcal{SC})$
24:             $\vec{u}(h) \leftarrow specificity(T, t, F, f)$
25: return $\vec{u}$
```

sense sharing the same hypernym (h in the algorithm) according to the WordNet taxonomy integrated in BabelNet (lines 4-6).

On all hyponym clusters we impose the restriction that they should have at least one lexicalization of the hypernym above the standard lexical specificity threshold 2 (lines 16-18). The reason why we include this in the unified representation is to reduce some noise detected by applying our previously proposed unified algorithm (Camacho-Collados et al., 2015c). Finally, if the cluster passes the threshold, the specificity is computed for the set of all the hyponyms of h, even those which do not occur in the sub-corpus $\mathcal{SC}$ (lines 20-24). As in Section 4.1.1, F and f denote the frequencies in the reference corpus $\mathcal{RC}$ (Wikipedia) and the sub-corpus $\mathcal{SC}$, respectively. In this case, the frequencies correspond to the aggregation of frequencies of h and all its hyponyms.

Our clustering of sibling words into a single cluster represented by their common hypernym transforms a lexical space into a unified semantic space. This space has multilingual synsets as dimensions, enabling their direct comparability across languages. We evaluated this feature of the unified vectors on the task of cross-lingual word similarity in Section 4.3.3. The clustering may also be viewed as an implicit disambiguation of potentially ambiguous words, as they are disambiguated into their intended sense represented by their hypernym, resulting in a more accurate semantic representation.

4.1.5 Vector comparison

As our vector comparison method for the lexical and unified vectors we use the square-rooted Absolute Weighted Overlap (Camacho-Collados et al., 2015b,c), which is based on the Weighted Overlap measure (Pilehvar et al., 2013). For notational brevity, we will refer to the square-rooted Absolute Weighted Overlap as Weighted Overlap (WO). WO compares two vectors on the basis of their overlapping dimensions, which are harmonically weighted by their absolute rankings. For this measure the vectors are viewed as *semantic sets* or *ranked lists* (Webber et al., 2010), as the weights are only used to sort the elements within the vector and their actual values are not used in the calculation. Formally, Weighted Overlap between two vectors $\vec{v_1}$ and $\vec{v_2}$ is defined as follows:

$$WO(\vec{v_1}, \vec{v_2}) = \sqrt{\frac{\sum_{d \in O} \left(rank(d, \vec{v_1}) + rank(d, \vec{v_2})\right)^{-1}}{\sum_{i=1}^{|O|} (2i)^{-1}}} \tag{4.10}$$

where O is the set of overlapping dimensions (i.e., concepts or words) between the two vectors and $rank(d, \vec{v_i})$ is the rank of dimension d in the vector $\vec{v_i}$. Absolute WO differs from the original WO, which takes into account the relative ranks of the dimensions with respect to the overlapping dimensions, instead of considering all the dimensions of the vector. Owing to the use of absolute ranks this measure gives lower scores in comparison to the original WO. This is the reason behind the use of the square-root operator, which smooths the distribution of values over the [0,1] scale. This metric has been shown to suit specificity-based vectors more than the conventional cosine distance (Camacho-Collados et al., 2015b).

In contrast, for comparing our embedded vector representations we use cosine, which is the usual measure used in the literature to measure similarity in an embedding space (Mikolov et al., 2013a; Chen et al., 2014; Li and Jurafsky, 2015). The dimensions of the embedded representations are not interpretable and the dimension values do not represent weights, thus rank-based WO is not applicable on this setting.

4.2 From a synset to its vector representations

In Section 4.1 we proposed three vector representations of an arbitrary text or subcorpus $\mathcal{SC}$ belonging to a larger collection. We now see how we leverage these representations to obtain a semantic vector representation for concepts and named entities. As knowledge base we use BabelNet[5], a multilingual encyclopedic dictionary which merges WordNet with other lexical and encyclopedic resources such as Wikipedia and Wiktionary, thanks to its use of an automatic mapping algorithm (Navigli and Ponzetto, 2012). We chose BabelNet due to its large coverage of named entities and concepts in hundreds of languages. Moreover, concepts and named entities are organized into a full-fledged taxonomy which integrates the WordNet taxonomy, which is the one used in our experiments, and, from its latest versions, the Wikipedia Bitaxonomy (Flati et al., 2014), WikiData, and *is-a* relations coming from open information extraction techniques (Delli Bovi et al., 2015b). Our approach

[5] See Section 2 for more information about BabelNet.

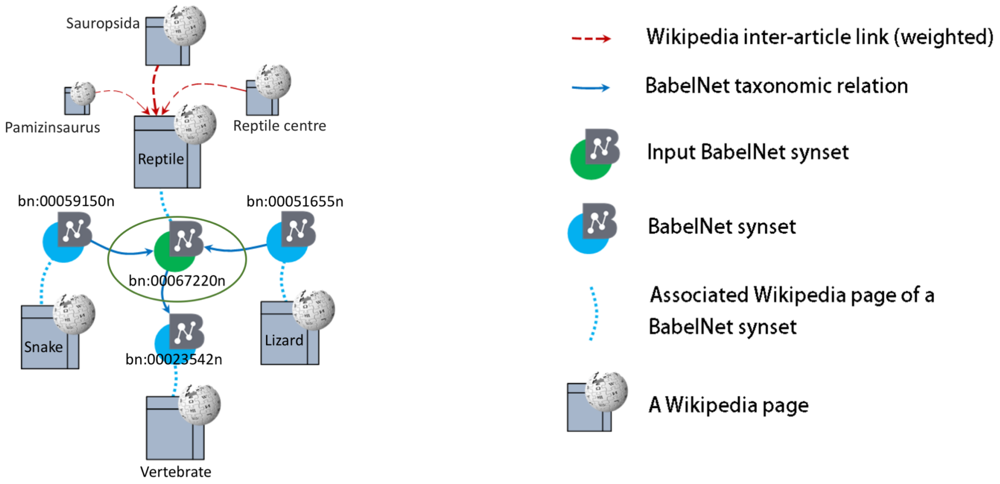


Figure 4.2. Our procedure for getting contextual information of the sample BabelNet synset represented by its main sense *reptile*^1_n.

makes use of the full power of BabelNet, as it exploits the complementary information of the distributional statistics in Wikipedia articles that are tied to the taxonomical relations in BabelNet. The rest of this section is divided into two parts. We first show how we collect contextual information for a given synset (Section 4.2.1) and then explain how this contextual information is processed in order to obtain our vector representations (Section 4.2.2).

4.2.1 Getting contextual information for a given synset

The goal of the first step is to create a subcorpus $\mathcal{SC}_s$ for a given BabelNet synset s. Let $\mathcal{W}_s$ be the set containing the Wikipedia page corresponding to the concept s (wp_s henceforth) and all the related Wikipedia pages that have an outgoing link to that page. Note that at this stage $\mathcal{W}_s$ might be empty if there is no Wikipedia page corresponding to the BabelNet synset s. We further enrich $\mathcal{W}_s$ by adding the corresponding Wikipedia pages of the hypernyms and hyponyms of s in the taxonomy of BabelNet. Figure 4.2 illustrates our procedure for obtaining contextual information. Let $\mathcal{SC}_s$ be the set of content words occurring in the Wikipedia pages of $\mathcal{W}_s$ after tokenization and lemmatization. The frequency of each content word w of $\mathcal{SC}_s$ is calculated as follows:

$$f(w) = \sum_{i=1}^{n} \lambda_i f_i(w) \tag{4.11}$$

where n is the number of Wikipedia pages in $\mathcal{W}_s$, $f_i(w)$ is the frequency of w in the Wikipedia page $p_i \in \mathcal{W}_s$ (i=1,...,n), and λ_i is the weight assigned to the page p_i to denote its importance. In the following subsection we explain how we calculate the weight λ_i for a given page p_i.

Weighting semantic relations

In this section we explain how we weight the BabelNet semantic relations (i.e., λ_i in Equation 4.11) between the target synset s and the i-th page in $\mathcal{W}_s$. In previous versions of Nasari (Camacho-Collados et al., 2015b,c) we were making an assumption that all the Wikipedia pages in $\mathcal{W}_s$ were equally important (i.e., $\lambda_i = 1, \forall i \leq n$). In this work we set more meaningful weights for these pages on the basis of the source and type of semantic connection to the target synset s.

A Wikipedia page in $\mathcal{W}_s$ may come from three different sources (see Section 4.2.1): (1) the Wikipedia page corresponding to s (wp_s), (2) the related Wikipedia pages that have an outgoing link to the page wp_s, and (3) the Wikipedia pages that are connected to s through taxonomic relations in BabelNet, We compute and assign a weight in the $[0, 1]$ range for the pages of each type as follows:

1. The **Wikipedia page corresponding to the BabelNet synset** s (i.e., wp_s) is assigned the highest possible weight of 1.

2. The weights for the **related Wikipedia pages that have an outgoing link to** wp_s are computed as follows. We first compute the lexical vectors of these Wikipedia pages, as well as for wp_s. We then apply Weighted Overlap (see Section 4.1.5) to calculate the similarity between the lexical vectors of each of these pages and that of wp_s. These similarity scores denote the weight of each related Wikipedia page. In order to reduce the high number of ingoing links in some cases, and to improve the quality of these links, we prune the ingoing links to include only the top 100 links on the basis of their similarity scores and those whose similarity score is higher than 0.25.

3. Given there is a possibility that a particular synset does not have a Wikipedia page associated with it, the Wikipedia pages coming from taxonomic relations cannot be calculated as in the previous case. In this case, the **Wikipedia pages coming from taxonomic relations** are given a fixed score of 0.85, which was calculated as follows. We picked a set of 100 random taxonomic relations and calculated the average similarity score among the 100 pairs by using our previous Nasari system.

4.2.2 Transforming the contextual information into vector representations

Once we have gathered a corpus $\mathcal{SC}_s$ for a given BabelNet synset s and computed the associated frequencies $f(w)$ for each word w in $\mathcal{SC}_s$, we proceed to calculate the lexical, embedded and unified vectors of s as explained in Sections 4.1.2, 4.1.3 and 4.1.4, respectively. In our experiments, we used the whole Wikipedia corpus as our reference corpus $\mathcal{RC}$ (Wikipedia dump of December 2014)[6]. We computed Nasari lexical and unified vectors for English, German, French, Italian, and Spanish. The number of synset vectors for each of these languages is, respectively, 4.42M, 1.51M, 1.48M, 1.10M and 1.07M. On average, for the English language, the contextual information of a synset is composed of a subcorpus $\mathcal{SC}_s$ of 1561 words in total coming

[6] Each language uses the Wikipedia corpus in its respective language as reference corpus.

Bank (financial institution)			Bank (geography)		
English	French	Spanish	English	French	Spanish
bank	banque	banco	river	eau	banco
banking	bancaire	bancario	stream	castor	limnología
deposit	crédit	banca	bank	berge	ecología
credit	financier	financiero	riparian	canal	barrera
money	postal	préstamo	creek	barrage	estuarios
loan	client	entidad	flow	zone	isla
commercial_bank	dépôt	déposito	water	perchlorate	interés
central_bank	billet	crédito	watershed	humide	laguna

Table 4.1. Top-weighted dimensions from the lexical vectors of the financial and geographical senses of *bank*.

Bank (financial institution)			Bank (geography)		
English	French	Spanish	English	French	Spanish
‡bank^2_n	‡banque^1_n	‡banco^1_n	⋆stream^1_n	eau^1_n	inclinación^9_n
reserve^2_n	•fonds^2_n	⋆institución_fin...^1_n	river^1_n	eau$^{15}_n$	lago^1_n
⋆financial_ins...^1_n	⋄dépôt^9_n	⋄depósito$^{15}_n$	‡body_of_water^1_n	excrément^1_n	‡cuerpo_de_agua^1_n
⋄deposit^8_n	∘emprunt^2_n	†finanzas^1_n	flow^1_n	castor^1_n	⋆arroyo^1_n
banking^2_n	paiement^1_n	•dinero^2_n	course^2_n	‡étendue_d'eau^1_n	tierra$^{11}_n$
†finance^1_n	argent^2_n	∘préstamo^2_n	bank^1_n	fourrure^1_n	costa^1_n

Table 4.2. Top-weighted dimensions from the unified vectors of the financial and geographical senses of *bank*. We represent each synset by one of its word senses. Word senses marked with the same symbol across languages correspond to the same BabelNet synset.

from 17 Wikipedia pages. For the embedded vectors, we took as word embeddings the pre-trained word and phrase vectors from Word2Vec[7]. These vectors were trained on a 100-billion English corpus from Google News and have 300 dimensions.

Lexical and unified synset vectors example. We show in Tables 4.1 and 4.2, respectively, the top-weighted dimensions of the lexical and unified vector representations for the financial and geographical senses of the noun *bank* in three different languages, i.e., English, French and Spanish.[8] As can be seen, the two senses of *bank* are clearly identified and distinguished from each other according to the top dimensions of their vectors, irrespective of their language and type. Additionally, note that the unified vectors are comparable across languages. We mark in Table 4.2, across different languages, those word senses that correspond to the same BabelNet synset. It can be seen from the Table that the unified vectors in different languages share many of their top elements.

[7] The pre-trained Word2Vec word embeddings were downloaded at `https://code.google.com/p/Word2Vec/`.

[8] In Table 4.2, *financial_ins...* refer to *financial_institution* and *institución_fin...* to *institución_financiera*.

Bank (financial institution)		**Bank (geography)**		***bank***	
Closest synsets	Cosine	Closest synsets	Cosine	Closest synsets	Cosine
Deposit account	0.99	Stream bed	0.98	Bank (financial ins...)	0.86
Universal bank	0.99	Current (stream)	0.97	Universal bank	0.86
British banking	0.98	River engineering	0.97	British banking	0.86
German banking	0.98	Braided river	0.97	German banking	0.85
Commercial bank	0.98	Fluvial terrace	0.97	Branch (banking)	0.85
Banking in Israel	0.98	Bar (river morphology)	0.97	McFadden Act	0.85
Financial institution	0.98	River	0.97	Four Northern Banks	0.84
Community bank	0.97	Perennial stream	0.96	State bank	0.84

Table 4.3. Closest embedded vectors from the BabelNet synsets corresponding to the financial and geographical senses of *bank*, and from the word *bank*.

Word and synset embeddings example. The dimensions are not interpretable in the embedded vectors. Therefore, a better way to distinguish different senses would be to show their closest elements in the space (using cosine as vector similarity measure). Table 4.3 shows the eight closest synsets to the word *bank*, as well as those closest to two specific senses of this word, i.e., the financial and geographical senses (recall that in our embedded vector representation words and synsets share the same space).[9] In this case, both senses of *bank* are again clearly distinguished by their closest BabelNet synsets in the space. Looking at the closest synsets to the word *bank* we can see that most of these are rather similar to the financial meaning of *bank*, with lower cosine values, though. This shows that the predominant sense of the word *bank* in the Google News corpus (on which the word embeddings are trained) is clearly its financial sense. We note that using our embedded vector representation one can easily compute the predominance of the senses of a word by directly comparing the representation of that word with those of its individual senses. Our shared space also provides a suitable framework for studying the ambiguity of words.

4.3 Intrinsic Evaluation: Semantic Similarity

Semantic similarity is the most popular benchmark for the intrinsic evaluation of different semantic representation techniques. The task here is to measure the semantic closeness of two linguistic items. The similarity of two items can be directly computed by comparing their corresponding vector representations. As we mentioned in Section 4.1.5, we opted for Weighted Overlap as our vector comparison method for lexical and unified representations, and cosine for the embedded representations. Note that by using our approach we obtain representations for individual BabelNet synsets. Moreover, because BabelNet merges different resources, our representations can be used to calculate the semantic similarity between any two semantic units within and across different resources, for instance between two Wikipedia pages, two WordNet synsets, or a Wikipedia page and a WordNet synset.

[9] In Table 4.3, *Bank (financial ins...)* refer to *Bank (financial institution).*

We benchmark our semantic similarity procedure on the word similarity task. Word similarity is a specific task from semantic similarity in which we measure how semantically close two words are. In order to be able to compute the similarity between words we first need to map the two words to their corresponding synsets. However, this mapping is a straightforward process, thanks to the multilingual sense inventory of BabelNet. As frequently done in this task, we measure the similarity between two words w and w' as the similarity between their closest senses (Resnik, 1995; Budanitsky and Hirst, 2006; Pilehvar et al., 2013):

$$sim(w, w') = \max_{\vec{v_1} \in \mathcal{L}_w, \vec{v_2} \in \mathcal{L}_{w'}} VC(\vec{v_1}, \vec{v_2}) \tag{4.12}$$

where $\mathcal{L}_w$ represents the set of synsets which contain w as one of its lexicalizations. As vector comparison VC we use WO (see Section 4.1.5) to compare lexical and unified representations, and cosine for the embedded representations.

Note that, thanks to our unified representation, w and w' may belong to different languages. Throughout this section on the tasks based on semantic similarity, **Nasari$_{\textbf{lexical}}$** and **Nasari$_{\textbf{unified}}$** represent the systems based on the lexical and unified vectors, respectively. We refer to the combination of both lexical and unified vectors as **Nasari**. This combination is based on the average similarity scores given by lexical and unified vectors for each sense pair. We also report results of our **Nasari$_{\textbf{embed}}$** vector representations which use the pre-trained Word2Vec vectors as input. We performed experiments on monolingual word similarity for English and other languages (presented in Sections 4.3.1 and 4.3.2, respectively) and cross-lingual similarity (presented in Section 4.3.3). Additionally, we evaluate our embedded representations in a cross-level semantic similarity task in Section 4.3.4.

4.3.1 Monolingual word similarity: English

Datasets The majority of benchmarks for word similarity are available only for the English language. We compare our approach with other state-of-the-art word similarity systems on standard English word similarity datasets. We chose the standard MC-30 (Miller and Charles, 1991), WordSim-353 (Finkelstein et al., 2002), and SimLex-999 (Hill et al., 2015) as evaluation benchmarks. **MC-30** consists of a subset of RG-65 (Rubenstein and Goodenough, 1965) which was re-annotated following new similarity guidelines. WordSim-353 consists of 353 word pairs, including both concepts and named entities. In the original WordSim-353 similarity conflated relatedness in the same dataset. In order to avoid this conflation, Agirre et al. (2009a) cleverly divided the dataset into two subsets: the first one concerned relatedness while the second subset focused on similarity, the latter being the one used in our experiments. We will refer to this similarity subset of 203 word pairs as **WS-Sim** henceforth. Finally, we took the noun pairs from the **SimLex-999** dataset as our last evaluation benchmark. The complete SimLex-999 dataset is composed of 999 word pairs, 666 of which are noun pairs.

Comparison systems We selected state-of-the-art approaches which are available online as comparison systems. These systems can be split into two categories: knowledge-based and corpus-based. As knowledge-based, we selected two approaches based on the WordNet semantic graph: (Pilehvar et al., 2013, **ADW**)[10] and (Lin, 1998, **Lin**)[11]. Another knowledge-based approach is (Gabrilovich and Markovitch, 2007, **ESA**)[12], which represents a word in a semantic space of Wikipedia articles. We also compared our systems with four corpus-based approaches[13]. Firstly, we took the pre-trained *word embeddings* of **Word2Vec** (Mikolov et al., 2013a)[14], the same used for our NASARI$_{\text{embed}}$ system (see Section 4.2.2). Then, we took the best predictive and count-based models for semantic similarity released by Baroni et al. (2014)[15]. The best predictive model is based on Word2Vec (**Word2Vec*** henceforth), while the best count-based models (**PMI-SVD***) are traditional co-occurrence vectors based on Pointwise Mutual Information (PMI) combined with a Singular Value Decomposition (SVD) dimensionality reduction. Finally, we benchmarked our system against two embedding-based sense representation approaches. The first approach, **Chen** henceforth (Chen et al., 2014), leverages word embeddings, WordNet glosses and a WSD system for creating sense embeddings[16]. The second one, called **SensEmbed** (Iacobacci et al., 2015), uses BabelNet as the main knowledge source and also relies on pre-disambiguated text by using a WSD system. We report the results of these last two methods when using the same closest senses strategy used by our systems.

Results Table 4.4 shows Pearson and Spearman correlation performance of our systems and all comparison systems on the three considered datasets[17]. Both lexical and unified vectors, especially the lexical ones, prove to be quite robust across datasets. The combination of both lexical and unified vectors does not show any noticeable improvement over the lexical vectors single-handed. Our system gets the highest average Pearson correlation among all systems, outperforming even the embedding-based approaches which use one dataset (SensEmbed) or two datasets (Word2Vec*) in order to tune their hyperparameters[18]. In terms of Spearman correlation, our system based on the lexical vectors also achieves the highest average performance among the systems which do not use any of the datasets for tuning with a single point advantage over Word2Vec. NASARI$_{\text{embed}}$ also proves to be quite competitive, outperforming all Word2Vec approaches in terms of Pearson correlation and obtaining the best overall result on MC-30.

Lin, which does not perform particularly well on MC-30 and WS-Sim, surprisingly obtains the best overall performance on the SimLex-999 dataset, which is largest

[10] ADW implementation available at `https://github.com/pilehvar/ADW`

[11] Results for Lin were obtained from the WS4J implementation available at `https://code.google.com/p/ws4j/`

[12] ESA implementation available at DKProSimilarity package (Bär et al., 2013).

[13] All the corpus-based approaches mentioned in the evaluation use cosine as comparison measure.

[14] The pre-trained models are available at `https://code.google.com/p/Word2Vec/`. They were trained on a Google News corpus of about 100 billion words.

[15] Both models were trained on a 2.8 billion-token corpus including the English Wikipedia. They are available at `clic.cimec.unitn.it/composes/semantic-vectors.html`

[16] The sense representations were downloaded from `http://pan.baidu.com/s/1eQcPK8i`

[17] Inter-annotator agreement (IAA) is also reported for those datasets for which this information is available. IAA is reported in terms of average pairwise Spearman correlation.

[18] Levy et al. (2015a) showed that with a fine tuning, Word2Vec can achieve a 0.79 Spearman correlation performance on WS-Sim, higher than the 0.77 Spearman correlation reported by Baroni et al. (2014) on that dataset.

	MC-30		WS-Sim		SimLex-999 (nouns)		Average	
	r	ρ	r	ρ	r	ρ	r	ρ
NASARI	0.89	0.78	0.74	0.72	0.50	0.49	**0.71**	0.67
NASARI$_{\text{lexical}}$	0.88	0.81	0.74	0.73	0.51	0.49	**0.71**	**0.68**
NASARI$_{\text{unified}}$	0.88	0.78	0.72	0.70	0.49	0.48	0.70	0.65
NASARI$_{\text{embed}}$	**0.91**	0.83	0.68	0.68	0.48	0.46	0.69	0.66
ESA	0.59	0.65	0.45	0.53	0.16	0.23	0.40	0.47
Lin	0.76	0.72	0.66	0.62	**0.58**	**0.58**	0.67	0.64
ADW	0.79	0.83	0.63	0.67	0.44	0.45	0.62	0.65
Chen	0.82	0.82	0.63	0.64	0.48	0.44	0.64	0.63
Word2Vec	0.80	0.80	**0.76**	**0.77**	0.46	0.45	0.67	0.67
Word2Vec*	$0.83^{\ddagger}$	$0.83^{\ddagger}$	$0.76^{\ddagger}$	$0.78^{\ddagger}$	0.48	0.49	0.69	0.70
PMI-SVD*	$0.76^{\ddagger}$	$0.71^{\ddagger}$	$0.68^{\ddagger}$	$0.66^{\ddagger}$	0.40	0.40	0.61	0.59
SensEmbed	0.89	**0.88**	0.65	0.75	$0.46^{\dagger}$	$0.47^{\dagger}$	0.67	0.70
IAA	-	-	-	$0.61^{\diamond}$	-	0.61		

Table 4.4. Pearson (r) and Spearman (ρ) correlations on RG-65, MC-30, WS-Sim and SimLex-999 (noun instances) datasets. We show the best performance obtained by Baroni et al. (2014) out of 48 configurations across different datasets including WS-Sim and RG-65 (highlighted by ‡). We show the SenseEmbed configuration tuned on the SimLex-999 dataset (highlighted by †). The inter-annotator agreement of the whole WordSim-353 (highlighted with ⋄) was reported to be 0.61, no inter-annotator agreement has been reported for the WS-Sim subset.

considered dataset, consisting of 666 noun pairs. Our system gets the second best overall performance on this dataset. A closer look at the output of the similarity scores given by our system compared to the gold standard shows noticeable errors when measuring the similarity between antonym pairs, which are heavily represented in this dataset. These antonym pairs were given consistently low values across the dataset, irrespective of the target words, whereas we argue that the similarity scores ought to vary according to the particular semantics of the antonym pairs. For instance, the pair *day-night* gets a score of 1.9 in the 0-10 scale, while our system gets a much higher 8.0 score[19]. A similar phenomenon is found on the *sunset-sunrise* pair. Nevertheless, in both cases the words in the pair belong to coordinate synsets in WordNet. In fact, recent works have shown how significant performance improvements can be obtained on this dataset by simply tweaking usual word embedding approaches to handle antonymy (see Section 5.4.1 for more details). This differs from the scores given in the WordSim-353 dataset, in which antonym pairs were considered as similar (Hill et al., 2015). It is outside the scope of this work to change this feature of our system in order to resolve its judgment differences with respect to the human annotation of antonym pairs in the SimLex-999 dataset.

[19] All scores have been converted to the 0-10 scale for this example.

4.3.2 Multilingual word similarity

Datasets We took the **RG-65** dataset as evaluation benchmark. The language of this dataset was originally English (Rubenstein and Goodenough, 1965). It was later translated into French (Joubarne and Inkpen, 2011), German (Gurevych, 2005) and Spanish (Camacho-Collados et al., 2015a). We used the four versions of the dataset for our experiments.

Comparison systems We benchmark our system against other multilingual word similarity approaches. **Wiki-wup** (Ponzetto and Strube, 2007) and **LSA-Wiki** (Granada et al., 2014) are systems which use Wikipedia as their main knowledge resource. We also provide results for co-occurrence-based methods such as **PMI** and **SOC-PMI** Joubarne and Inkpen (2011) and for newer word embedding techniques (Mikolov et al., 2013a; Faruqui et al., 2015). For word embeddings we report results for the **Word2Vec** model[20] and for an approach retrofitting these Word2Vec vectors into WordNet (**Retrofitting**) (Faruqui et al., 2015). For the Spanish language no result was reported in Faruqui et al. (2015) for Word2Vec, so we trained Word2Vec with the same hyperparameters of **Word2Vec*** Baroni et al. (2014) on the *Spanish Billion Words Corpus*[21] (Cardellino, 2016). We used these Spanish word embeddings as input for our NASARI$_{\text{embed}}$ system in this language. Additionally, we report results for pre-trained embeddings in all four languages (Al-Rfou et al., 2013, **Polyglot**)[22]. These vectors have sixty-four dimensions and were trained on the Wikipedia corpus. We also compare this system with our embedded representations of synsets by using the *polyglot* word embeddings as input continuous representations (see Section 4.1.3). We will refer to this latter method as NASARI$_{\text{polyglot}}$.

Results Table 4.5 shows Pearson and Spearman correlation performance of our systems and all comparison systems on the RG-65 word similarity datasets for English, French, German and Spanish[23]. Our system outperforms all multilingual comparison systems in English, French and German in terms of both Pearson and Spearman correlation. For the Spanish language our system surprisingly slightly outperforms the human inter-annotator agreement (which was calculated in terms of average pairwise Pearson correlation), hence demonstrating the competitiveness of our approach in this language too.

The Polyglot-embed multilingual representations do not show a particular potential for the task. The reason behind these results may be due, apart from the inherent ambiguity of words, to their low dimensionality (64) and small vocabulary (100K words). However, our embedded representation using these word embeddings (NASARI$_{\text{polyglot}}$) hugely improves the original vectors (obtaining an

[20] For English, the pre-trained models of Word2Vec trained on a Google News corpus of 100 billion words were considered for the evaluation. For French and German, a corpus of a 1 billion tokens from Wikipedia was used for training.

[21] Downoloaded from `http://crscardellino.me/SBWCE/`

[22] The pre-trained polyglot word representations were downloaded from `https://sites.google.com/site/rmyeid/projects/polyglot`.

[23] Inter-annotator agreement (IAA) is also reported for the languages for which this information is available. IAA is reported in terms of average pairwise Pearson correlation.

English	r	ρ	**French**	r	ρ	**German**	r	ρ	**Spanish**	r	ρ
NASARI	0.81	0.78	NASARI	**0.82**	0.73	NASARI	0.69	0.65	NASARI	**0.85**	0.79
NASARI$_{\text{lexical}}$	0.80	0.78	NASARI$_{\text{lexical}}$	0.80	0.70	NASARI$_{\text{lexical}}$	0.69	0.67	NASARI$_{\text{lexical}}$	**0.85**	0.79
NASARI$_{\text{unified}}$	0.80	0.76	NASARI$_{\text{unified}}$	**0.82**	**0.76**	NASARI$_{\text{unified}}$	**0.71**	**0.68**	NASARI$_{\text{unified}}$	0.82	0.77
NASARI$_{\text{embed}}$	**0.82**	**0.80**	–	–	–	–	–	–	NASARI$_{\text{embed}}$	0.79	0.77
SOC-PMI	0.61	–	SOC-PMI	0.19	–	SOC-PMI	0.27	–	–	–	–
PMI	0.41	–	PMI	0.34	–	PMI	0.40	–	–	–	–
LSA-Wiki	0.65	0.69	LSA-Wiki	0.57	0.52	–	–	–	–	–	–
Wiki-wup	0.59	–	–	–	–	Wiki-wup	0.65	–	–	–	–
Word2Vec	–	0.73	Word2Vec	–	0.47	Word2Vec	–	0.53	Word2Vec*	0.80	**0.80**
Retrofitting	–	0.77	Retrofitting	–	0.61	Retrofitting	–	0.60	–	–	–
NASARI$_{\text{polyglot}}$	0.74	0.77	NASARI$_{\text{polyglot}}$	0.60	0.69	NASARI$_{\text{polyglot}}$	0.46	0.52	NASARI$_{\text{polyglot}}$	0.68	0.74
Polyglot	0.51	0.55	Polyglot	0.38	0.35	Polyglot	0.18	0.15	Polyglot	0.51	0.56
IAA	0.85$^{\diamond}$	-	IAA	-	-	IAA	0.81	-	IAA	0.83	-

Table 4.5. Pearson (r) and Spearman (ρ) correlation performance of different systems on the English, French, German and Spanish RG-65 datasets. The inter-annotator of the English RG-65 (highlighted with $\diamond$) was calculated for a subset of fifteen annotators.

average twenty-three Pearson and twenty-eight Spearman correlation points improvement). NASARI$_{\text{polyglot}}$, despite achieving lower results than our three representations, achieves competitive results with respect to other comparison systems, with the added benefit of being applicable to many languages (pre-trained polyglot embeddings are available for more than a hundred languages).

4.3.3 Cross-lingual word similarity

Datasets We have used the RG-65 cross-lingual datasets released by Camacho-Collados et al. (2015a) for English, French, German and Spanish. These datasets[24] were automatically constructed by taking the manually-curated multilingual RG-65 datasets from the previous section as input. In total, we evaluated on six datasets consisting of all the possible language pair combinations for the four languages. Each dataset consists of a set cross-lingual word pairs, ranging from 95 to 125 pairs.

Comparison systems As cross-lingual comparison systems, we have included the best results provided by the **CL-MSR-2.0** system (Kennedy and Hirst, 2012). This system applies PMI on an English-French parallel corpus obtained from WordNet. Additionally, we provide results for some of the best performing systems in English word similarity by using English as a pivot language[25]. Baseline pivot systems include the WordNet-based system **ADW** (Pilehvar et al., 2013), the pre-trained **Word2Vec** word embeddings Mikolov et al. (2013a) and the top performing Word2Vec model in similarity obtained by Baroni et al. (2014) (**Word2Vec***), and the best count-based model obtained by Baroni et al. (2014) (**PMI-SVD***). See Section 4.3.1 for more details on these comparison systems. We also report results for our system using

[24] The cross-lingual datasets are available at `http://lcl.uniroma1.it/similarity-datasets/`

[25] Non-English words are translated by using Google Translate.

Measure	**EN-FR**		**EN-DE**		**EN-ES**		**FR-DE**		**FR-ES**		**DE-ES**		**Average**	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
NASARI$_{\text{unified}}$	**0.84**	0.79	**0.79**	0.79	**0.84**	0.82	0.75	0.70	**0.86**	0.78	**0.81**	**0.80**	**0.82**	0.78
CL-MSR-2.0	0.30	–	–	–	–	–	–	–	–	–	–	–	–	–
NASARI$_{pivot}$	0.79	0.69	0.78	0.76	0.80	0.74	**0.79**	0.70	0.80	0.67	0.72	0.68	0.78	0.71
ADW$_{pivot}$	0.80	0.82	0.73	**0.82**	0.78	**0.84**	0.72	**0.77**	0.81	**0.81**	0.68	0.72	0.75	**0.80**
Word2Vec$_{pivot}$	0.77	0.82	0.70	0.73	0.76	0.80	0.65	0.70	0.75	0.76	0.64	0.63	0.71	0.74
Word2Vec*$_{pivot}$	0.75	**0.84**	0.69	0.76	0.75	0.82	0.77	0.73	0.74	0.79	0.64	0.64	0.72	0.76
PMI-SVD*$_{pivot}$	0.76	0.76	0.72	0.74	0.77	0.77	0.65	0.69	0.76	0.74	0.62	0.61	0.71	0.72

Table 4.6. Pearson (r) and Spearman (ρ) correlation performances of different similarity measures on the six cross-lingual RG-65 datasets. Notation: English (EN), French (FR), German (DE), Spanish (ES).

the combination of lexical and unified English NASARI vectors. We refer to all these systems using English as pivot language as *pivot*.

Results Table 4.6 shows cross-lingual word similarity results according to Pearson and Spearman correlation performance. In this section we only report results for our unified vector representations, as their dimensions are BabelNet synsets, which are multilingual and therefore may be used for direct cross-lingual comparison. Our unified vector representations outperform all comparison systems (both types) in terms of Pearson correlation performance except for the French-German pair, in which our *pivot* system obtains the best result. It is interesting to note that our English monolingual similarity proves to be the most robust across language pairs among all *pivot* systems according to Pearson correlation measure, demonstrating the reliability of our system also on a purely monolingual scheme. *Pivot* systems prove to be competitive, outperforming the only cross-lingual baseline which does not use a pivot language. In fact, despite obtaining relatively modest Pearson results, ADW obtains the best results according to the Spearman correlation measure (our unified vector representations obtain the second best result overall). In terms of the harmonic mean of Pearson and Spearman, used as official measure in a previous semantic similarity SemEval task (Jurgens et al., 2014) and in previous works (Hassan and Mihalcea, 2011), our system outperforms ADW (second overall system) by three points (0.80 to 0.77), demonstrating the effectiveness of our direct cross-lingual word comparison with respect to the use of English as a pivot language.

4.3.4 Cross-level semantic similarity

Finally, we evaluated our embedded representations on the word to sense semantic similarity task. Recall from Section 4.2.2 that our embedded vector representations share the same space with word embeddings. Therefore, in order to calculate the similarity between a word and a sense, we only have to compute the cosine similarity between their respective vector representations.

Dataset As our benchmark we opted for the *Word to Sense* (word2sense) similarity subtask of the **SemEval-2014 Cross-Level Semantic Similarity** (CLSS) task

	r	ρ
NASARI$_{\text{embed}}$	0.40	0.40
Meerkat Mafia	**0.44**	**0.44**
SemantiKLUE	0.39	0.39
SimCompass	0.31	0.33

Table 4.7. Pearson and Spearman correlation performance of different systems on the *word2sense* test set of SemEval-2014 task on Cross-Level Semantic Similarity.

(Jurgens et al., 2014). The subtask provides 500 word-sense pairs for its test dataset. Each pair is associated with a score denoting the semantic overlap between the two items. From the dataset we took the subset in which the senses are noun instances[26] (277 pairs). This dataset includes many words that are not usually integrated in a knowledge source, such as slang words. Our embedded representations are particularly suitable for this task as they can handle BabelNet Out-Of-Vocabulary words thanks to the shared space of words and senses: if a word is not integrated in BabelNet sense inventory, we simply use the word embedding sharing the same surface form of the given sense.

Comparison systems Thirty-eight systems participated in the *word2sense* subtask. We compare the performance of our embedded representations with the three best performing participating systems in this subtask. **Meerkat Mafia** (Kashyap et al., 2014) is a system that relies on Latent Semantic Analysis (LSA) and uses external dictionaries to handle OOV words. **SemantiKLUE** (Proisl et al., 2014) combines a set of different unsupervised and supervised techniques to measure semantic similarity. The third system, the most similar to our system, is **SimCompass** (Mihalcea and Wiebe, 2014), which relies on deep learning word embeddings and uses WordNet as its only knowledge source.

Results Table 4.7 shows Pearson and Spearman correlation performance of the NASARI system with embedded representations together with the three comparison systems. Meerkat Mafia obtains the best overall performance on this dataset. Our system is the second best system, outperforming the remaining 37 participating systems of the SemEval task. Interestingly, NASARI$_{\text{embed}}$ provides a considerable improvement over SimCompass (0.09 and 0.07 in terms of Pearson and Spearman correlations, respectively), which is also based on word embeddings and uses WordNet as lexical resource.

[26] Note that our embedded representations can be used to measure the similarity between words with any Part Of Speech tag.

4.4 Analysis

In order to gain a better insight into the role some of the key components of our system's pipeline play in the overall performance, we carried out an ablation test. In particular, we were interested in evaluating the impact and importance of the following three components:

1. **Lexical specificity.** To check how lexical specificity (see Section 4.1.1) fares against the standard *tf-idf* measure (Jones, 1972), we generated Nasari lexical vectors in which weights were calculated using the conventional *tf-idf.* Given a word w, we calculate $TFidf(w)$ as follows:

$$TFidf(w) = f(w) \log \frac{|D|}{|\{p \in D : w \in p\}|} \tag{4.13}$$

   where $f(w)$ is the frequency of w in the subcorpus $\mathcal{SC}_s$ representing the contextual information of the synset s (see Section 4.2.1) and D is the set of all pages in Wikipedia. We computed two sets of *tf-idf*-based lexical vectors. The first version, called Nasari-TFidf, keeps all the dimensions in the vector. For the second version, Nasari-TFidf-3000d, we follow Gong et al. (2014) and prune the vector to its top 3000 non-zero dimensions. This pruning is similar to the one performed automatically by lexical specificity, which reduces the number of non-zero dimensions while retaining the interpretability of the vector dimensions.

2. **Weighted semantic relations.** To assess the advantage we gain from introducing weights to semantic relations (see Section 4.2.1), we computed a version of our lexical vectors in which the semantic relations were uniformly weighted (i.e., $\lambda_i = 1, \forall i \in \{1, \ldots, n\}$ in Equation 4.11), as was the case in our earlier work Camacho-Collados et al. (2015c). We will refer to this version as Nasari-unif.weight.

3. **Combination strategy of embeddings.** Finally, we carried out an analysis to compare the harmonic combination of word embeddings (see Section 4.1.3) against uniform combination (i.e., averaging). For this purpose, we computed the embedding vector for a given synset as the centroid of all the embeddings of the words present in its corresponding lexical vector. We will refer to this variant as Nasari-av.embed in our tables.

These three components were analyzed intrinsically on the word similarity task. The whole pipeline of Nasari was left unchanged for all variants, except for these components mentioned above.

Table 4.8 shows the results of the ablation test on word similarity[27]. Our default Nasari$_{\text{lexical}}$ system consistently outperforms all baselines in all datasets of both tasks, demonstrating the reliability of the proposed lexical specificity and the preweighting of the semantic relations. This result is especially meaningful taking

[27] In Camacho-Collados et al. (2016c) we performed an additional ablation test on the sense clustering task, leading to similar conclusions.

	Word Similarity					
	MC-30		WS-Sim		SimLex (nouns)	
	r	ρ	r	ρ	r	ρ
Nasari$_{\text{lexical}}$	**0.88**	**0.81**	**0.74**	**0.73**	**0.51**	**0.49**
Nasari-TFidf	0.84	0.77	0.71	0.71	0.46	0.46
Nasari-TFidf-3000d	0.85	0.79	0.72	0.72	0.48	0.47
Nasari-unif.weight	0.86	0.79	0.73	0.72	0.49	0.48
Nasari$_{\text{embed}}$	**0.91**	**0.83**	**0.68**	**0.68**	**0.48**	**0.46**
Nasari-av.embed	0.81	0.75	0.58	0.63	0.40	0.41

Table 4.8. Ablation test. Pearson (r) and Spearman (ρ) correlations on RG-65, MC-30, WS-Sim and SimLex-999 (noun instances) word similarity datasets.

into account that our default system is the one with the fewest non-zero dimensions on average among the four evaluated approaches. In fact, the average number of non-zero dimensions of our Nasari$_{\text{lexical}}$ vectors was 162, which is lower than the 280 non-zero dimensions of Nasari-unif.weight, 1033 of Nasari-TFidf-3000d[28], and 1561 of Nasari-TFidf. This low average number of non-zero dimensions enables a fast processing of the vectors, i.e., they are computationally faster to work with.

As far as the Nasari$_{\text{embed}}$ vectors are concerned, our default system consistently obtained significantly better results when compared to the baseline (Nasari-av.embed). In general, Nasari-av.embed produces consistently high similarity values, even for non-similar pairs. This is due to the fact that words that are not very relevant to the input synset (i.e., relatively low lexical specificity values) are given the same weight as words that are clearly more relevant (i.e., high lexical specificity values). This, in turn, is why a weighted average of the word embeddings in the lexical vector leads to more accurate results than a simple average.

4.5 Conclusion

In this chapter we presented Nasari, a novel technique for the representation of concepts and named entities in arbitrary languages. Our approach combines the structural knowledge from semantic networks with the statistical information derived from text corpora for effective representation of millions of BabelNet synsets, including WordNet nominal synsets and all Wikipedia pages. We evaluated our representations intrinsically on the semantic similarity benchmark, reporting state-of-the-art performance on several datasets across these tasks and in different languages. Additionally, we also devised robust frameworks that enable direct application of our representations into different tasks (presented in subsequent chapters), namely word sense disambiguation (Chapter 6), sense clustering (Section 7.1), domain labeling (Section 7.2) and text classification (Chapter 8).

[28] In Nasari-TFidf-3000d the maximum number of non-zero dimensions is set to 3000, but in many cases the vector has actually a lower number of non-zero dimensions.

Three type of sense representation were put forward: two explicit vector representations (unified and lexical) in which vector dimensions are interpretable and a latent embedding-based representation. Each representation has its own advantages and disadvantages. In general, a combination of lexical and unified vectors led to the most reliable results in the semantic similarity experiments (Sections 4.3). Among the three representations, the lexical representation (i.e., NASARI$_{\text{lexical}}$) obtained the best performance in monolingual settings. However, although the lexical vectors are sparse and computationally easy to work with in many applications, the dimensionality is high as it is equal to the vocabulary size. In contrast, our embedded representation (i.e., NASARI$_{\text{embed}}$) has a fixed low number of latent dimensions. Additionally, embedded synset vectors share the same space with the word embeddings used as input. As regards our unified representation (i.e., NASARI$_{\text{unified}}$), not only does it provide an effective way for representing word senses in different languages, but, thanks to its unified semantic space, it also enables a direct comparison of different representations across languages. In addition to being multilingual, NASARI improves over the existing techniques by providing a high coverage for millions of concepts and named entities defined in the BabelNet sense inventory.

Chapter 5

SW2V: Senses and Words to Vectors

Recently, approaches based on neural networks which embed words into low-dimensional vector spaces from text corpora (i.e. word embeddings) have become increasingly popular (Mikolov et al., 2013a; Pennington et al., 2014). Word embeddings have proved to be beneficial in many Natural Language Processing tasks, such as Machine Translation (Zou et al., 2013), syntactic parsing (Weiss et al., 2015), and Question Answering (Bordes et al., 2014), to name a few. Despite their success in capturing semantic properties of words, these representations are generally hampered by an important limitation: the inability to discriminate among different meanings of the same word.

As mentioned in Chapter 3, previous works have addressed this limitation by automatically inducing word senses from monolingual corpora (Schütze, 1998; Reisinger and Mooney, 2010; Huang et al., 2012; Neelakantan et al., 2014; Tian et al., 2014; Li and Jurafsky, 2015; Vu and Parker, 2016; Qiu et al., 2016), or bilingual parallel data (Guo et al., 2014; Ettinger et al., 2016; Šuster et al., 2016). However, these approaches learn solely on the basis of statistics extracted from text corpora and do not exploit knowledge from semantic networks. Additionally, their induced senses are neither readily interpretable (Panchenko et al., 2017) nor easily mappable to lexical resources, which limits their application. Recent approaches have utilized semantic networks to inject knowledge into existing word representations (Yu and Dredze, 2014; Faruqui et al., 2015; Goikoetxea et al., 2015; Speer and Lowry-Duda, 2017; Mrkšic et al., 2017), but without solving the meaning conflation issue. In order to obtain a representation for each sense of a word, a number of approaches have leveraged lexical resources to learn sense embeddings as a result of post-processing conventional word embeddings (Chen et al., 2014; Johansson and Pina, 2015; Jauhar et al., 2015; Rothe and Schütze, 2015; Pilehvar and Collier, 2016).

Instead, we propose SW2V (*Senses and Words to Vectors*), a neural model that exploits knowledge from both text corpora and semantic networks in order to simultaneously learn embeddings for both words and senses. In contrast to Nasari (Chapter 4), this model is not focused on learning only nominal senses (concepts or entities), but also verbs, adjectives or adverbs. Moreover, our model provides three additional key features: (1) both word and sense embeddings are represented in the same vector space, (2) it is flexible, as it can be applied to different predictive models, and (3) it is scalable for very large semantic networks and text corpora.

5.1 Connecting words and senses in context

In order to jointly produce embeddings for words and senses, SW2V needs as input a corpus where words are connected to senses[1] in each given context. One option for obtaining such connections could be to take a sense-annotated corpus as input. However, manually annotating large amounts of data is extremely expensive and therefore impractical in normal settings. Obtaining sense-annotated data from current off-the-shelf disambiguation and entity linking systems is possible, but generally suffers from two major problems. First, supervised systems are hampered by the very same problem of needing large amounts of sense-annotated data. Second, the relatively slow speed of current disambiguation systems, such as graph-based approaches (Hoffart et al., 2012; Agirre et al., 2014; Moro et al., 2014), or word-expert supervised systems (Zhong and Ng, 2010; Iacobacci et al., 2016; Melamud et al., 2016), could become an obstacle when applied to large corpora.

This is the reason why we propose a simple yet effective unsupervised *shallow word-sense connectivity* algorithm, which can be applied to virtually any given semantic network and is linear on the corpus size. The main idea of the algorithm is to exploit the connections of a semantic network by associating words with the senses that are most connected within the sentence, according to the underlying network.

Shallow word-sense connectivity algorithm. Formally, a corpus and a semantic network are taken as input and a set of connected words and senses is produced as output. We define a semantic network as a graph (S, E) where the set S contains synsets (nodes) and E represents a set of semantically connected synset pairs (edges). Algorithm 2 describes how to connect words and senses in a given text (sentence or paragraph) T. First, we gather in a set S_T all candidate synsets of the words (including multiwords up to trigrams) in T (lines 1 to 3). Second, for each candidate synset s we calculate the number of synsets which are connected with s in the semantic network and are included in S_T, excluding connections of synsets which only appear as candidates of the same word (lines 5 to 10). Finally, each word is associated with its top candidate synset(s) according to its/their number of connections in context, provided that its/their number of connections exceeds a threshold $\theta = \frac{|S_T|+|T|}{2\,\delta}$ (lines 11 to 17).[2] This parameter aims to retain relevant connectivity across senses, as only senses above the threshold will be connected to words in the output corpus. θ is proportional to the reciprocal of a parameter δ,[3]

[1] In this work we focus on senses only, but other items connected to words may be used as well (e.g. supersenses or images).

[2] As mentioned above, all unigrams, bigrams and trigrams present in the semantic network are considered. In the case of overlapping instances, the selection of the final instance is performed in this order: mention whose synset is more connected (i.e. n is higher), longer mention and from left to right.

[3] Higher values of δ lead to higher recall, while lower values of δ increase precision but lower the recall. We set the value of δ to 100, as it was shown to produce a fine balance between precision and recall. This parameter may also be tuned on downstream tasks.

Algorithm 2 Shallow word-sense connectivity

```
Input: Semantic network $(S, E)$ and text $T$ represented as a bag of words
Output: Set of connected words and senses $T^* \subset T \times S$
Set of synsets $S_T \leftarrow \emptyset$
for each word $w \in T$
    $S_T \leftarrow S_T \cup S_w$ ($S_w$: set of candidate synsets of $w$)
Minimum connections threshold $\theta \leftarrow \frac{|S_T|+|T|}{2\,\delta}$
Output set of connections $T^* \leftarrow \emptyset$
for each $w \in T$
    Relative maximum connections $max = 0$
    Set of senses associated with $w$, $C_w \leftarrow \emptyset$
    for each candidate synset $s \in S_w$
        Number of edges $n = |s' \in S_T : (s, s') \in E \ \& \ \exists w' \in T : w' \neq w \ \& \ s' \in S_{w'}|$
        if $n \geq max \ \& \ n \geq \theta$ then
            if $n > max$ then
                $C_w \leftarrow \{(w, s)\}$
                $max \leftarrow n$
            else
                $C_w \leftarrow C_w \cup \{(w, s)\}$
    $T^* \leftarrow T^* \cup C_w$
return Output set of connected words and senses $T^*$
```

and directly proportional to the average text length and number of candidate synsets within the text.

The complexity of the proposed algorithm is $N + (N \times \alpha)$, where N is the number of words of the training corpus and α is the average polysemy degree of a word in the corpus according to the input semantic network. Considering that non-content words are not taken into account (i.e. polysemy degree 0) and that the average polysemy degree of words in current lexical resources (e.g. WordNet or BabelNet) does not exceed a small constant (e.g. 3) in any language, we can safely assume that the algorithm is linear in the size of the training corpus. Hence, the training time is not significantly increased in comparison to training words only, irrespective of the corpus size and the predictive model. This enables a fast training on large amounts of text corpora, in contrast to current unsupervised disambiguation algorithms. Additionally, as we will show in Section 5.3.2, this algorithm does not only speed up significantly the training phase, but also leads to more accurate results.

Note that with our algorithm a word is allowed to have more than one sense associated. In fact, current lexical resources like WordNet or BabelNet are hampered by the high granularity of their sense inventories (Hovy et al., 2013). In Section 7.1 we show how our sense embeddings are particularly suited to deal with this issue.

5.2 Joint training of words and senses

The goal of our approach is to obtain a shared vector space of words and senses. To this end, our model extends conventional word embedding models by integrating explicit knowledge into its architecture. While we will focus on the Continuous Bag Of Words (CBOW) architecture of Word2Vec (Mikolov et al., 2013a), our extension

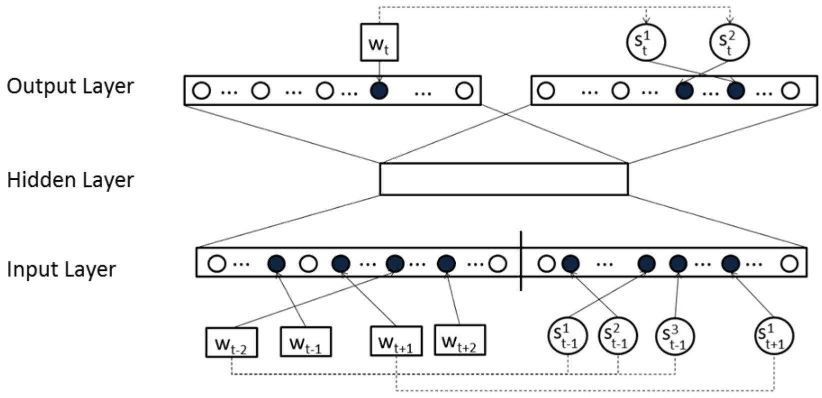


Figure 5.1. The SW2V architecture on a sample training instance using four context words. Dotted lines represent the virtual link between words and associated senses in context. In this example, the input layer consists of a context of two previous words (w_{t-2}, w_{t-1}) and two subsequent words (w_{t+1}, w_{t+2}) with respect to the target word w_t. Two words (w_{t-1}, w_{t+2}) do not have senses associated in context, while w_{t-2}, w_{t+1} have three senses (s^1_{t-1}, s^2_{t-1}, s^3_{t-1}) and one sense associated (s^1_{t+1}) in context, respectively. The output layer consists of the target word w_t, which has two senses associated (s^1_t, s^2_t) in context.

can easily be applied similarly to Skip-Gram, or to other predictive approaches based on neural networks. The CBOW architecture is based on the feedforward neural network language model (Bengio et al., 2003) and aims at predicting the current word using its surrounding context. The architecture consists of input, hidden and output layers. The input layer has the size of the word vocabulary and encodes the context as a combination of one-hot vector representations of surrounding words of a given target word. The output layer has the same size as the input layer and contains a one-hot vector of the target word during the training phase.

Our model extends the input and output layers of the neural network with word senses[4] by exploiting the intrinsic relationship between words and senses. The leading principle is that, since a word is the surface form of an underlying sense, updating the embedding of the word should produce a consequent update to the embedding representing that particular sense, and vice-versa. As a consequence of the algorithm described in the previous section, each word in the corpus may be connected with zero, one or more senses. We refer to the set of senses connected to a given word within the specific context as its *associated senses*.

Formally, we define a training instance as a sequence of words (being w_t the target word) $W = w_{t-n}, ..., w_t, ..., w_{t+n}$ and senses $S = S_{t-n}, ..., S_t, ..., S_{t+n}$, where $S_i = s^1_i, ..., s^{k_i}_i$ is the sequence of all associated senses in context of $w_i \in W$. Note that S_i might be empty if the word w_i does not have any associated sense. In our model each target word takes as context both its surrounding words and all the senses associated with them. In contrast to the original CBOW architecture, where the training criterion is to correctly classify w_t, our approach aims to predict the

[4] Our model can also produce a space of words and synset embeddings as output: the only difference is that all synonym senses would be considered to be the same item, i.e. a synset.

word w_t and its set S_t of associated senses. This is equivalent to minimizing the following loss function:

$$E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S_t} \log(p(s|W^t, S^t))$$

where $W^t = w_{t-n}, ..., w_{t-1}, w_{t+1}, ..., w_{t+n}$ and $S^t = S_{t-n}, ..., S_{t-1}, S_{t+1}, ..., S_{t+n}$. Figure 5.1 shows the organization of the input and the output layers on a sample training instance. In what follows we present a set of variants of the model on the output and the input layers.

5.2.1 Output layer alternatives

Both words and senses. This is the default case explained above. If a word has one or more associated senses, these senses are also used as target on a separate output layer.

Only words. In this case we exclude senses as target. There is a single output layer with the size of the word vocabulary as in the original CBOW model.

Only senses. In contrast, this alternative excludes words, using only senses as target. In this case, if a word does not have any associated sense, it is not used as target instance.

5.2.2 Input layer alternatives

Both words and senses. Words and their associated senses are included in the input layer and contribute to the hidden state. Both words and senses are updated as a consequence of the backpropagation algorithm.

Only words. In this alternative only the surrounding words contribute to the hidden state, i.e. the target word/sense (depending on the alternative of the output layer) is predicted only from word features. The update of an input word is propagated to the embeddings of its associated senses, if any. In other words, despite not being included in the input layer, senses still receive the same gradient of the associated input word, through a virtual connection. This configuration, coupled with the only-words output layer configuration, corresponds exactly to the default CBOW architecture of Word2Vec with the only addition of the update step for senses.

Only senses. Words are excluded from the input layer and the target is predicted only from the senses associated with the surrounding words. The weights of the words are updated through the updates of the associated senses, in contrast to the only-words alternative.

5.3 Analysis of Model Components

In this section we analyze the different components of SW2V, including the nine model configurations (Section 5.3.1) and the algorithm which generates the connections between words and senses in context (Section 5.3.2). In what follows we describe the common analysis setting:

- **Training model and hyperparameters.** For evaluation purposes, we use the CBOW model of Word2Vec with standard hyperparameters: the dimensionality of the vectors is set to 300 and the window size to 8, and hierarchical softmax is used for normalization. These hyperparameter values are set across all experiments.

- **Corpus and semantic network.** We use a 300M-words corpus from the UMBC project (Han et al., 2013), which contains English paragraphs extracted from the web.[5] As semantic network we use BabelNet. We chose BabelNet owing to its wide coverage of named entities and lexicographic knowledge (see Section 2.3).

- **Benchmark.** Word similarity has been one of the most popular benchmarks for *in-vitro* evaluation of vector space models (Pennington et al., 2014; Levy et al., 2015a). For the analysis we use two word similarity datasets: the similarity portion (Agirre et al., 2009a, WS-Sim) of the WordSim-353 dataset (Finkelstein et al., 2002) and RG-65 (Rubenstein and Goodenough, 1965). In order to compute the similarity of two words using our sense embeddings, similarly to what was performed with Nasari (see Section 4.3), we apply the standard closest senses strategy, using cosine similarity (cos) as comparison measure between senses:

$$sim(w_1, w_2) = \max_{s \in S_{w_1}, s' \in S_{w_2}} \cos(\vec{s}_1, \vec{s}_2) \tag{5.1}$$

  where S_{w_i} represents the set of all candidate senses of w_i and $\vec{s}_i$ refers to the sense vector representation of the sense s_i.

5.3.1 Model configurations

In this section we analyze the different configurations of our model in respect of the input and the output layer on a word similarity experiment. Recall from Section 5.2 that our model could have words, senses or both in either the input and output layers. Table 5.1 shows the results of all nine configurations on the WS-Sim and RG-65 datasets.

As shown in Table 5.1, the best configuration according to both Spearman and Pearson correlation measures is the configuration which has only senses in the input layer and both words and senses in the output layer.[6] In fact, taking only senses as input seems to be consistently the best alternative for the input layer. Our hunch is that the knowledge learned from both the co-occurrence information and the

[5] `http://ebiquity.umbc.edu/resource/html/id/351/UMBC-webbase-corpus`

[6] In this analysis we used the word similarity task for optimizing the sense embeddings, without caring about the performance of word embeddings or their interconnectivity. Therefore, this configuration may not be optimal for word embeddings and may be further tuned on specific applications. More information about different configurations in the documentation of the source code.

		Output											
		Words				**Senses**				**Both**			
		WS-Sim		**RG-65**		**WS-Sim**		**RG-65**		**WS-Sim**		**RG-65**	
		r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Input	**Words**	0.49	0.48	0.65	0.66	0.56	0.56	0.67	0.67	0.54	0.53	0.66	0.65
	Senses	0.69	0.69	0.70	0.71	0.69	0.70	0.70	**0.74**	**0.72**	**0.71**	**0.71**	**0.74**
	Both	0.60	0.65	0.67	0.70	0.62	0.65	0.66	0.67	0.65	**0.71**	0.68	0.70

Table 5.1. Pearson (r) and Spearman (ρ) correlation performance of the nine configurations of SW2V

semantic network is more balanced with this input setting. For instance, in the case of including both words and senses in the input layer, the co-occurrence information learned by the network would be duplicated for both words and senses.

5.3.2 Disambiguation / Shallow word-sense connectivity algorithm

In this section we evaluate the impact of our *shallow word-sense connectivity algorithm* (Section 5.1) by testing our model directly taking a pre-disambiguated text as input. In this case the network exploits the connections between each word and its disambiguated sense in context. For this comparison we used Babelfy[7] (Moro et al., 2014), a state-of-the-art graph-based disambiguation and entity linking system based on BabelNet. We compare to both the default Babelfy system which uses the *Most Common Sense* (MCS) heuristic as a back-off strategy and, following Iacobacci et al. (2015), we also include a version in which only instances above the Babelfy default confidence threshold are disambiguated (i.e. the MCS back-off strategy is disabled). We will refer to this latter version as Babelfy* and report the best configuration of each strategy according to our analysis.

Table 5.2 shows the results of our model using the three different strategies on RG-65 and WS-Sim. Our shallow word-sense connectivity algorithm achieves the best overall results. We believe that these results are due to the semantic connectivity ensured by our algorithm and to the possibility of associating words with more than one sense, which seems beneficial for training, making it more robust to possible disambiguation errors and to the sense granularity issue (Erk et al., 2013). The results are especially significant considering that our algorithm took a tenth of the time needed by Babelfy to process the corpus.

5.4 Intrinsic Evaluation

We perform a qualitative and quantitative evaluation of important features of SW2V in three different tasks. First, in order to compare our model against standard word-based approaches, we evaluate our system quantitatively in the word similarity task (Section 5.4.1) and estimate the coherence of our unified vector space by measuring the interconnectivity of word and sense embeddings (Section 5.4.2).

[7] `http://babelfy.org/`

	WS-Sim		**RG-65**	
	r	ρ	r	ρ
Shallow	**0.72**	**0.71**	**0.71**	**0.74**
Babelfy	0.65	0.63	0.69	0.70
Babelfy*	0.63	0.61	0.65	0.64

Table 5.2. Pearson (r) and Spearman (ρ) correlation performance of SW2V integrating our *shallow* word-sense connectivity algorithm (default), Babelfy, or Babelfy*.

Experimental setting. Throughout all the experiments we use the same standard hyperparameters mentioned in Section 5.3 for both the original Word2Vec implementation and our proposed model SW2V. For SW2V we use the same optimal configuration according to the analysis of the previous section (only senses as input, and both words and senses as output) for all tasks. As training corpus we take the full 3B-words UMBC webbase corpus and the Wikipedia (Wikipedia dump of November 2014), used by three of the comparison systems. We use BabelNet 3.0 (SW2V$_{\text{BN}}$) and WordNet 3.0 (SW2V$_{\text{WN}}$) as semantic networks.

Comparison systems. As comparison systems, we include three state-of-the-art pre-trained knowledge-based sense vector representations:

- Chen et al. (2014)[8] developed a sense representation model based on a number of steps: pre-trained word embeddings, sense vector initialization based on WordNet glosses, corpus disambiguation and joint training of words and senses.
- Iacobacci et al. (2015)[9] developed SensEmbed, a sense embedding model based on BabelNet. It consists of running the CBOW model of Word2Vec over the Wikipedia corpus in which the words are replaced with BabelNet senses by using the off-the-shelf Babelfy disambiguation system (Moro et al., 2014).
- Rothe and Schütze (2015)[10] developed AutoExtend, an extension of word embeddings for WordNet senses and synsets. AutoExtend takes pre-trained word embeddings as input (Word2Vec in their original model and our experiments) and propagates them to WordNet sense and synset embeddings by using an autoencoding framework based on a set of constraints over the word embeddings.

5.4.1 Word similarity

In this section we evaluate our sense representations on the standard SimLex-999 (Hill et al., 2015) and MEN Bruni et al. (2014) word similarity datasets[11]. SimLex

[8] `http://pan.baidu.com/s/1eQcPK8i`

[9] `http://lcl.uniroma1.it/sensembed/`

[10] We used the AutoExtend code (`http://cistern.cis.lmu.de/~sascha/AutoExtend/`) to obtain sense vectors using W2V embeddings trained on UMBC (GoogleNews corpus used in their pre-trained models is not publicly available). We also tried the code to include BabelNet as lexical resource, but it was not easily scalable (BabelNet is two orders of magnitude larger than WordNet).

[11] To enable a fair comparison we did not perform experiments on the small datasets used in Section 5.3 for validation.

			SimLex-999		MEN	
	System	**Corpus**	r	ρ	r	ρ
Senses	SW2V$_{\text{BN}}$	UMBC	**0.49**	**0.47**	0.75	0.75
	SW2V$_{\text{WN}}$	UMBC	0.46	0.45	**0.76**	**0.76**
	AutoExtend	UMBC	0.47	0.45	0.74	0.75
	AutoExtend	Google-News	0.46	0.46	0.68	0.70
	SW2V$_{\text{BN}}$	Wikipedia	0.47	0.43	0.71	0.73
	SW2V$_{\text{WN}}$	Wikipedia	0.47	0.43	0.71	0.72
	SensEmbed	Wikipedia	0.43	0.39	0.65	0.70
	Chen et al. (2014)	Wikipedia	0.46	0.43	0.62	0.62
Words	Word2vec	UMBC	0.39	0.39	0.75	0.75
	Retrofitting$_{\text{BN}}$	UMBC	0.47	0.46	0.75	**0.76**
	Retrofitting$_{\text{WN}}$	UMBC	0.47	0.46	**0.76**	**0.76**
	Word2vec	Wikipedia	0.39	0.38	0.71	0.72
	Retrofitting$_{\text{BN}}$	Wikipedia	0.35	0.32	0.66	0.66
	Retrofitting$_{\text{WN}}$	Wikipedia	0.47	0.44	0.73	0.73

Table 5.3. Pearson (r) and Spearman (ρ) correlation performance on the SimLex-999 and MEN word similarity datasets.

and MEN contain 999 and 3000 word pairs, respectively, which constitute, to our knowledge, the two largest similarity datasets comprising a balanced set of noun, verb and adjective instances. As explained in Section 5.3, we use the closest sense strategy for the word similarity measurement of our model and all sense-based comparison systems. As regards the word embedding models, words are directly compared by using cosine similarity. We also include a *retrofitted* version of the original Word2Vec word vectors (Faruqui et al., 2015, Retrofitting[12]) using WordNet (Retrofitting$_{\text{WN}}$) and BabelNet (Retrofitting$_{\text{BN}}$) as lexical resources.

Table 5.3 shows the results of SW2V and all comparison models in SimLex and MEN. SW2V consistently outperforms all sense-based comparison systems using the same corpus, and clearly performs better than the original Word2Vec trained on the same corpus. Retrofitting decreases the performance of the original Word2Vec on the Wikipedia corpus using BabelNet as lexical resource, but significantly improves the original word vectors on the UMBC corpus, obtaining comparable results to our approach. However, while our approach provides a shared space of words and senses, Retrofitting still conflates different meanings of a word into the same vector.

Additionally, we noticed that most of the score divergences between our system and the gold standard scores in SimLex-999 were produced on antonym pairs, which are over-represented in this dataset: 38 word pairs hold a clear antonymy relation (e.g. *encourage-discourage* or *long-short*), while 41 additional pairs hold some degree of antonymy (e.g. *new-ancient* or *man-woman*).[13] In contrast to the consistently low gold similarity scores given to antonym pairs, our system varies its similarity

[12] `https://github.com/mfaruqui/retrofitting`

[13] Two annotators decided the degree of antonymy between word pairs: *clear antonyms*, *weak antonyms* or *neither*.

scores depending on the specific nature of the pair[14]. Recent works have managed to obtain significant improvements by tweaking usual word embedding approaches into providing low similarity scores for antonym pairs (Pham et al., 2015; Schwartz et al., 2015; Nguyen et al., 2016; Mrkšic et al., 2017), but this is outside the scope of our work.

5.4.2 Word and sense interconnectivity

In the previous experiment we evaluated the effectiveness of the sense embeddings on the word similarity task. In contrast, this experiment aims at testing the interconnectivity between word and sense embeddings in the vector space. As explained in Section 3.2, there have been previous approaches building a shared space of word and sense embeddings, but to date little research has focused on testing the semantic coherence of the vector space. To this end, we evaluate our model on a Word Sense Disambiguation (WSD) task, using our shared vector space of words and senses to obtain a *Most Common Sense* (MCS) baseline. The insight behind this experiment is that a semantically coherent shared space of words and senses should be able to build a relatively strong baseline for the task, as the MCS of a given word should be closer to the word vector than any other sense. The MCS baseline is generally integrated into the pipeline of state-of-the-art WSD and Entity Linking systems as a back-off strategy (Jin et al., 2009; Zhong and Ng, 2010; Moro et al., 2014; Raganato et al., 2017) and is used in various NLP applications (Bennett et al., 2016). Therefore, a system which automatically identifies the MCS of words from non-annotated text may be quite valuable, especially for resource-poor languages or large knowledge resources for which obtaining sense-annotated corpora is extremely expensive. Moreover, even in a resource like WordNet for which sense-annotated data is available (Miller et al., 1993, SemCor), 61% of its polysemous lemmas have no sense annotations (Bennett et al., 2016).

Given an input word w, we compute the cosine similarity between w and all its candidate senses, picking the sense leading to the highest similarity:

$$MCS(w) = \operatorname*{argmax}_{s \in S_w} \cos(\vec{w}, \vec{s}) \tag{5.2}$$

where $\cos(\vec{w}, \vec{s})$ refers to the cosine similarity between the embeddings of w and s. In order to assess the reliability of SW2V against previous models using WordNet as sense inventory, we test our model on the all-words SemEval-2007 (task 17) (Pradhan et al., 2007) and SemEval-2013 (task 12) (Navigli et al., 2013) WSD datasets. Note that our model using BabelNet as semantic network has a far larger coverage than just WordNet and may additionally be used for Wikification (Mihalcea and Csomai, 2007) and Entity Linking tasks. Since the versions of WordNet vary across datasets and comparison systems, we decided to evaluate the systems on the portion of the datasets covered by all comparison systems[15] (less than 10% of instances were removed from each dataset).

[14] For instance, the pairs *sunset-sunrise* and *day-night* are given, respectively, 1.88 and 2.47 gold scores in the 0-10 scale, while our model gives them a higher similarity score. In fact, both pairs appear as coordinate synsets in WordNet.

[15] We were unable to obtain the word embeddings of Chen et al. (2014) for comparison even after contacting the authors.

	SemEval-07	**SemEval-13**
SW2V	**39.9**	**54.0**
AutoExtend	17.6	31.0
Baseline	24.8	34.9

Table 5.4. F-Measure percentage of different MCS strategies on the SemEval-2007 and SemEval-2013 WSD datasets.

company$_n^2$ (military unit)		*school$_n^7$ (group of fish)*	
AutoExtend	**SW2V**	**AutoExtend**	**SW2V**
company$_n^9$	battalion$_n^1$	school	schools$_n^7$
company	battalion	school$_n^4$	sharks$_n^1$
company$_n^8$	regiment$_n^1$	school$_n^6$	sharks
company$_n^6$	detachment$_n^4$	school$_v^1$	shoals$_n^3$
company$_n^7$	platoon$_n^1$	school$_n^3$	fish$_n^1$
company$_v^1$	brigade$_n^1$	elementary	dolphins$_n^1$
firm	regiment	schools	pods$_n^3$
business$_n^1$	corps$_n^1$	elementary$_a^3$	eels
firm$_n^2$	brigade	school$_n^5$	dolphins
company$_n^1$	platoon	elementary$_a^1$	whales$_n^2$

Table 5.5. Ten closest word and sense embeddings to the senses *company$_n^2$* (military unit) and *school$_n^7$* (group of fish).

Table 5.4 shows the results of our system and AutoExtend on the SemEval-2007 and SemEval-2013 WSD datasets. SW2V provides the best MCS results in both datasets. In general, AutoExtend does not accurately capture the predominant sense of a word and performs worse than a baseline that selects the intended sense randomly from the set of all possible senses of the target word.

In fact, AutoExtend tends to create clusters which include a word and all its possible senses. As an example, Table 5.5 shows the closest word and sense embeddings of our SW2V model and AutoExtend to the *military* and *fish* senses of, respectively, *company* and *school*. AutoExtend creates clusters with all the senses of *company* and *school* and their related instances, even if they belong to different domains (e.g., firm$_n^2$ or business$_n^1$ clearly concern the *business* sense of *company)*. Instead, SW2V creates a semantic cluster of word and sense embeddings which are semantically close to the corresponding *company$_n^2$* and *school$_n^7$* senses.

5.5 Conclusion

In this chapter we proposed SW2V (*Senses and Words to Vectors*), a neural model which learns vector representations for words and senses in a joint training phase by exploiting both text corpora and knowledge from semantic networks. Data (including the preprocessed corpora and pre-trained embeddings used in the evaluation) and source code to apply our extension of the Word2Vec architecture to learn word and sense embeddings from any preprocessed corpus are freely available at `http://lcl.uniroma1.it/sw2v`. Unlike previous sense-based models which require post-processing steps and use WordNet as sense inventory, our model achieves a semantically coherent vector space of both words and senses as an emerging feature of a single training phase and is easily scalable to larger semantic networks like BabelNet. Finally, we showed, both quantitatively and qualitatively, some of the advantages of using our approach as against previous state-of-the-art word- and sense-based models in various tasks, and highlighted interesting semantic properties of the resulting unified vector space of word and sense embeddings. In Chapter 7 we explain how to make use of our SW2V embeddings in two tasks: sense clustering (Section 7.1) and collocation discovery (Section 7.4).

PART 2: Applications

In the first part we provided an overview of semantic representations of word senses, concepts and entities, and proposed two new models. We view these knowledge-based semantic representations as a bridge between lexical resources and NLP applications. We argue that the use of these sense representations could potentially lead to mutual benefits for improving and enriching lexical resources as well as in downstream NLP applications. In order to assess the reliability and flexibility of sense representations (and in particular the representations presented in Chapters 4 and 5) across different tasks, we carry out a comprehensive set of evaluations for various relevant applications. In particular, three complementary tasks are considered: Word Sense Disambiguation (Chapter 6), knowledge base enrichment (Chapter 7) and text classification (Chapter 8). A brief overview of the experiments across the four tasks follows:

- **Word Sense Disambiguation** (Chapter 6). In this chapter we present experiments on WSD by presenting simple monolingual and multilingual knowledge-based frameworks based on Nasari. We present results using various knowledge resources and languages as sense inventory for monolingual WSD in Section 6.2. In Section 6.3 we present a multilingual system which exploits parallel or comparable corpora for producing high-quality sense annotations in arbitrary languages using distributional semantic similarity via Nasari$_{\text{embed}}$.

- **Knowledge Base Enrichment** (Chapter 7). In this chapter we focus on applications which are aimed at improving the quality of knowledge resources. In particular, in this chapter we present the following four applications:

  - **Sense Clustering** (Section 7.1). Current sense inventories are hampered by the high granularity of their sense inventories (Hovy et al., 2013; Pilehvar et al., 2017). In this section we show how sense representations (and in particular Nasari and SW2V) are specially suited to deal with this issue.
  - **Domain Labeling** (Section 7.2). In this section we present an approach which exploits the content and the structural properties of knowledge resources to provide a unified domain annotation. The approach exploits the topical similarity of Nasari along with

    various graph-based heuristics. Thanks to the use of this technique we released BabelDomains (Camacho-Collados and Navigli, 2017), a unified resource providing domain labels for over 2.6M lexical items from BabelNet, Wikipedia and WordNet.

  - **Hypernym Discovery** (Section 7.3): Hypernymy relations constitute the backbone of lexical taxonomies and are used in different NLP applications (Prager et al., 2008; Yahya et al., 2013; Hoffart et al., 2014). In this work we first leverage the domain annotations constructed in the previous section for splitting training data into domains of knowledge. This domain clustering leads to consistent improvements in hypernym discovery when integrated into a supervised distributional model based on sense embeddings.
  - **Collocation Discovery** (Section 7.4): WordNet contains a wide set of relations, ranging from hypernymy to meronymy or holonymy. In this section we attempt to extend WordNet with fine-grained collocational information, exploiting a supervised distributional model exploiting knowledge-based sense embeddings models, including SW2V. This results on a new resource, named ColWordNet (Espinosa-Anke et al., 2016b), which extends WordNet with collocational information at different levels.

- **Downstream NLP Applications** (Chapter 8). In this chapter we investigate the integration of word senses into downstream applications, in particular text categorization and sentiment analysis. To this end we modify the input of a state-of-the-art neural network architecture for text classification by taking into account senses and pre-trained sense embeddings based on Nasari. Our analysis shows the potential of this integration of senses and its advantages with respect to mainstream word-based models.

Chapter 6

Word Sense Disambiguation

Word Sense Disambiguation (WSD) is a core task in natural language understanding. Given a target word in context, the task consists of associating it with an entry from a pre-defined sense repository (Navigli, 2009). WSD may eventually be applied to any Natural Language Processing task, enabling an understanding of the sentences by the machine which is not usually achieved by mainstream statistical approaches, and could benefit applications such as Machine Translation (Vickrey et al., 2005; Xiong and Zhang, 2014; Liu et al., 2017), sentiment analysis (Flekova and Gurevych, 2016; Pilehvar et al., 2017) or Information Retrieval (Schütze and Pedersen, 1995; Zhong and Ng, 2012), to name but a few.

Sense inventories. One of the main knowledge sense repositories for WSD is WordNet (Navigli et al., 2013; Pradhan et al., 2007), which usually leads to a fine-grained type of disambiguation given the nature of the sense distinctions in WordNet. Another resource more recently used for this task is Wikipedia (Mihalcea and Csomai, 2007; Dandala et al., 2013b; Navigli et al., 2013), due to its wide coverage of named entities and multilinguality. Finally, a newer resource used as a knowledge repository that is gaining popularity thanks to its multilinguality and large coverage is BabelNet (Navigli et al., 2013; Moro and Navigli, 2015; Weissenborn et al., 2015a), which is also a merger of WordNet and Wikipedia, among other resources (see Chapter 2). Given the nature of our NASARI vectors (see Section 4), and in contrast to other WSD systems, we can therefore seamlessly disambiguate in any of these resources.

The remainder of the chapter is organized as follows. In Section 6.2 we present a framework for monolingual WSD based on the NASARI lexical vectors, which is applied to different languages and sense inventories. Then, in Section 6.3 we present a pipeline exploiting comparable or parallel corpora for jointly performing multilingual disambiguation coupling graph and semantic similarity cues using NASARI.

6.1 Related Work

Depending on their nature, WSD systems are divided into two main groups: supervised and knowledge-based. In what follows we summarize the current state of the art of these two types of approach.

6.1.1 Supervised WSD

Supervised models train different features extracted from manually sense-annotated corpora. These features have been mostly based on the information provided by the surroundings words of the target word (Keok and Ng, 2002; Navigli, 2009) and its collocations. Recently, more complex features based on word embeddings trained on unlabeled corpora have also been explored (Taghipour and Ng, 2015b; Rothe and Schütze, 2015; Iacobacci et al., 2016). These features are generally taken as input to train a linear classifier (Zhong and Ng, 2010; Shen et al., 2013). In addition to these conventional approaches, the latest developments in neural language models have motivated some researchers to include them in their WSD architectures (Kågebäck and Salomonsson, 2016; Melamud et al., 2016; Yuan et al., 2016).

Supervised models have traditionally been able to outperform knowledge-based systems Navigli (2009). However, obtaining sense-annotated corpora is highly expensive, and in many cases such corpora are not available for specific domains. In fact, the performance of supervised systems very much depends on the availability of sense-annotated data for the target word sense (Pilehvar and Navigli, 2014b). Hence, the applicability of these systems is limited to those words and languages for which such data is available, practically restricting them to a small subset of word senses and mainly for the English language only. This is the reason why some of these supervised methods have started to rely on unlabeled corpora as well. These approaches, which are often classified as *semi-supervised*, are targeted at overcoming the knowledge acquisition bottleneck of conventional supervised models. In fact, there is a line of research specifically aimed at automatically obtaining large amounts of high-quality sense-annotated corpora (Taghipour and Ng, 2015a; Raganato et al., 2016; Delli Bovi et al., 2017). These automatically obtained data have been exploited by semi-supervised models, which have been shown to outperform fully supervised systems in various settings (Taghipour and Ng, 2015b; Başkaya and Jurgens, 2016; Iacobacci et al., 2016; Yuan et al., 2016). In Section 6.3 we specifically study this problem and proposed automatic methods which exploit multilinguality for construct high-quality sense-annotated corpora. Additionally, we show how knowledge-based and supervised models may be complementary, opening up exciting new lines for future work (Section 6.2.4).

6.1.2 Knowledge-based WSD

In contrast to supervised systems, knowledge-based WSD techniques do not require any sense-annotated corpus. Instead, these approaches rely on the structure or content of manually-curated knowledge resources for disambiguation. One of the first approaches of this kind was Lesk (1986), which in its original version consisted of calculating the overlap between the context of the target word and its definitions as given by the sense inventory. Based on the same principle, various works have adapted the original algorithm by also taking into account definitions from related words (Banerjee and Pedersen, 2003), or by calculating the distributional similarity between definitions and the context of the target word (Basile et al., 2014; Chen et al., 2014). In addition to these approaches exploiting definition signals, an important branch of knowledge-based systems found their techniques on the structural properties of semantic graphs from lexical resources (Sinha and Mihalcea, 2007; Guo and Diab, 2010; Ponzetto and Navigli, 2010; Agirre et al., 2014; Moro et al., 2014; Weissenborn et al., 2015a; Tripodi and Pelillo, 2017). Generally, these graph-based WSD systems first create a graph representation of the input text and then exploit different graph-based algorithms over the given representation (e.g., PageRank) to perform WSD.

In contrast, we propose a knowledge-based WSD system based on distributional similarity, and more concretely on Nasari vectors (see Chapter 4). To this end, we put forward a flexible framework which can be used arbitrary languages and resources (Section 6.2) and another method exploiting at best comparable and parallel corpora for high-quality disambiguation and entity linking (Section 6.3).

6.2 Monolingual Word Sense Disambiguation

In this section we present a WSD framework in which we exploit the Nasari lexical vectors for monolingual WSD. Our framework is an improvement of the method presented in Camacho-Collados et al. (2015c). This method consisted of computing overlap between the Nasari lexical vector and the context of a target word, harmonically weighting the ranks of the overlapping words in the lexical vector. While this method proved effective for the WSD task, it still considered each word in context to be equally important (same weight) in the disambiguation process. For this work we present an approach which keeps to the spirit of this knowledge-based model based on Nasari vectors, but proposing an improvement on which words in context receive individual weights.

Methodology. Given a set of target words in a text $\mathcal{T}$, we build a lexical vector for the context, as explained in Section 4.1.2. Then, for each target word w in the text $\mathcal{T}$, we retrieve the set of all the possible BabelNet synsets which have this target word as one of its lexicalizations, a set we refer to as $\mathcal{L}_w$. Finally, we simply compute Weighted Overlap (see Section 4.1.5) between $\vec{v}_{lex}(\mathcal{T})$ (the lexical vector of the text $\mathcal{T}$) and the Nasari vectors corresponding to the BabelNet synsets that contain senses of w. In our setting, the top BabelNet synset in terms of WO score ($\hat{s}$) is selected as the best sense of the given target word:

$$\hat{s} = \operatorname*{argmax}_{s \in \mathcal{L}_w} WO(\vec{v}_{lex}(\mathcal{T}), \overrightarrow{\textsc{Nasari}}_{lex}(s)) \tag{6.1}$$

Experimental setting. We perform Word Sense Disambiguation experiments using two sense inventories: Wikipedia and WordNet. Recall from Section 2 that, since our main knowledge sense inventory is BabelNet, we can seamlessly disambiguate instances using either of these two knowledge sources. The setting of the system is the same in both cases, with only one difference: we use only BabelNet synsets[1] which are mapped to Wikipedia page or WordNet synset when disambiguating with

[1] In order to avoid disambiguating with synsets which are rarely used in practise and are isolated in the BabelNet graph, throughout all the experiments we only considered those BabelNet synsets with at least thirty edges in the BabelNet graph.

either of these resources, respectively. As has often been done in the literature (Vasilescu et al., 2004; Zhong and Ng, 2010; Moro et al., 2014), we use a back-off strategy to the Most Frequent Sense (MFS) baseline in the cases when our system does not provide a confident answer. Hence, in our WSD framework, we only tagged those instances whose top similarity score is higher than a given threshold θ. In order to compute θ, we use the English Wikipedia trial dataset provided within the SemEval 2013 WSD task (Navigli et al., 2013). The top performing value of θ was 0.20, value that is used across all WSD experiments[2].

Section 6.2.1 presents multilingual WSD experiments using Wikipedia as main sense inventory (a task that is strongly related to the *Wikification* task (Mihalcea and Csomai, 2007)), Section 6.2.2 presents experiments for the Named Entity Disambiguation task using BabelNet as sense inventory, and finally Section 6.2.3 presents the WSD results for English using WordNet as sense inventory.

6.2.1 Word Sense Disambiguation using Wikipedia

We used the **SemEval-2013 all-words WSD** dataset (Navigli et al., 2013) as benchmark for our multilingual evaluations[3]. This dataset includes texts for five different languages (English, French, German, Italian and Spanish) with an average of 1303 disambiguated instances per language, including multiword expressions and named entities.

Comparison systems. As comparison system we include **Babelfy** (Moro et al., 2014), a state-of-the-art graph-based system for multilingual joint WSD and Entity Linking. Babelfy relies on random walks in the BabelNet semantic network combined with various graph-based heuristics. We also report results for the best run on every language of the top SemEval-2013 system (Gutiérrez et al., 2013, **UMCC-DLSI**). As baseline, although difficult to beat in some WSD tasks (Navigli, 2009), we include the Most Frequent Sense (**MFS**[4]) heuristic. Finally, we report results from **Muffin** (Camacho-Collados et al., 2015c), our previous WSD system based on the NASARI vectors that, in contrast, used a WSD framework in which words in context were considered equally important.

Results. Table 6.1 shows F-Measure percentage results for our system and all comparison systems on the SemEval 2013 dataset. As we can see from the table,

[2] We considered values of θ from 0 to 1 with a step size of 0.05.

[3] In our experiments we used the Wikipedia dump of December 2014, as opposed to the one used in the original SemEval 2013 dataset. A few Wikipedia page titles had been updated since the creation of the dataset, so we had to update these titles in the gold standard too. For instance, the English Wikipedia page titled *Seven-day week* in the SemEval 2013 dataset has been updated in Wikipedia and is currently titled simply *Week*. Note that the Wikipedia page titles are the unique identifiers for a Wikipedia page, hence a change in a Wikipedia page title automatically modifies this unique identifier.

[4] MFS was provided as baseline by the task organizers. However, the MFS score for French was fixed with respect to Camacho-Collados et al. (2015c), which showed a lower MFS F-Measure score. The scorer provided by the organizers was case-sensitive whereas a few Wikipedia page titles in the gold standard file did not match the casing of those in the baseline file, which were all lowercased. This led to misalignments between the gold standard and the baseline file.

System	English	French	Italian	German	Spanish	Average
NASARI$_{\text{lexical}}$	86.3	**76.2**	83.7	**83.2**	82.9	**82.5**
MUFFIN	84.5	71.4	81.9	83.1	**85.1**	81.2
Babelfy	**87.4**	71.6	**84.3**	81.6	83.8	81.7
UMCC-DLSI	54.8	60.5	58.3	61.0	58.1	58.5
MFS	80.2	74.9	82.2	83.0	82.1	79.3

Table 6.1. F-Measure percentage performance on the SemEval-2013 Multilingual WSD datasets using Wikipedia as sense inventory.

although our system only achieves state-of-the-art results for French and German, it does achieve the best average performance among all languages, demonstrating its robustness across languages and outperforming the current state-of-the-art results of Babelfy. Our system outpeforms our previous WSD approach MUFFIN by over a point on average, highlighting our improvements on this particular WSD task for which we proposed a new framework.

6.2.2 Named Entity Disambiguation using BabelNet

In order to evaluate the quality of our named entity representation, we performed experiments on the Named Entity Disambiguation (NED) task. Given that NASARI provides semantic representations for both concepts and named entities, this task was analogous to WSD with the only difference being that in this task we only considered entity synsets as candidates. To this end, we used the English named entity dataset from the **All-Words Sense Disambiguation and Entity Linking SemEval 2015** task (Moro and Navigli, 2015). This dataset consists of 85 named entities to disambiguate.

Comparison systems. We benchmarked our disambiguation system against the SemEval 2015 top three performing systems, which were the only ones outperforming the MFS baseline: **DFKI** (Weissenborn et al., 2015b), **SUDOKU** (Manion, 2015), and **el92** (Ruiz and Poibeau, 2015). DFKI is a multi-objective system based on both global unsupervised and local supervised objectives. SUDOKU uses the Personalized PageRank algorithm after disambiguating monosemous instances within the text. Finally, el92 is based on a weighted voting of various disambiguation systems: Wikipedia Miner (Milne and Witten, 2008), TagME (Ferragina and Scaiella, 2010), DBpedia Spotlight (Mendes et al., 2011), and Babelfy (Moro et al., 2014).

Results. Table 6.2 shows F-Measure percentage results on the Named Entity portion of the SemEval 2015 WSD dataset[5]. Our system obtains the second overall position of all seventeen systems that participated in the SemEval 2015 Named Entity Disambiguation task. The combination of global unsupervised and local supervised

[5] We found an inaccuracy in an instance of the gold standard dataset. The unambiguous instance *KAlgebra* is disambiguated with the *KAlgebra* concept in the Catalan language, which belongs to a separate synset of the general *KAlgebra* concept in all languages. This instance is repeated nine times within the dataset. By fixing this issue, our system achieves F-Measure results of over 90%.

System	Type	F-Measure
NASARI$_{\text{lexical}}$	unsupervised	87.1
DFKI	supervised	**88.9**
SUDOKU	unsupervised	87.0
el92	systems mix	86.1
MFS	–	85.7

Table 6.2. F-Measure percentage performance on the English Named Entity Disambiguation dataset from the Multilingual All-Words Sense Disambiguation and Entity Linking SemEval 2015 task using BabelNet as sense inventory.

objectives of DFKI obtains the best overall results. As we show in Section 6.2.3 and discuss in Section 6.2.4, our system, based solely on global semantic features, generally improves when including local supervised features.

6.2.3 Word Sense Disambiguation using WordNet

For the task of English WSD using WordNet as main sense inventory, we used two recent SemEval WSD datasets: fine-grained all-words **SemEval-2007** (Pradhan et al., 2007) and all-words **SemEval-2013** (Navigli et al., 2013). We performed experiments on the 162 noun instances of the SemEval-2007 dataset. SemEval-2013's dataset contains 1644 instances.

Comparison systems. We include the state-of-the-art **IMS** system (Zhong and Ng, 2010) as a supervised system. As unsupervised systems, we report the performance of two graph-based approaches that are based on random walks over their respective semantic networks: BabelNet (Moro et al., 2014, **Babelfy**) and WordNet (Agirre and Soroa, 2009, **UKB**). Another approach that uses BabelNet as reference knowledge base is **Multi-Objective** (Weissenborn et al., 2015a) which views WSD as a multi-objective optimization problem. We also report the results of the best configuration of the top-performing system in the SemEval-2013 dataset, namely **UMCC-DLSI** (Gutiérrez et al., 2013). As in Section 6.2.1, we also include our earlier WSD system **Muffin** for comparison. Finally, we include a system called **Nasari+IMS**, which is based on our WSD framework with the only difference being that in this system we back-off to IMS instead of MFS[6].

Results. Table 6.3 shows the F-Measure percentage performance of all systems on the SemEval-2007 and SemEval-2013 WSD datasets. Similarly to the WSD results using Wikipedia as main sense inventory (Section 6.2.1), our system NASARI outperforms our previous MUFFIN system. NASARI in its default setting backing-off to MFS is only surpassed by Multi-Objective in SemEval-2013 and IMS in SemEval-2017, outperforming the remaining systems in both datasets.

Our system backing-off to IMS (NASARI+IMS) improves our default NASARI system in both datasets, obtaining the best performance among all systems on the

[6] The MFS baseline was obtained from the SemCor sense-annotated corpus (Miller et al., 1993).

System	SemEval-2013	SemEval-2007
NASARI$_{\text{lexical}}$	66.7	66.7
NASARI$_{\text{lexical}}$ +IMS	67.0	**68.5**
MUFFIN	66.0	66.0
Babelfy	65.9	62.7
UKB	62.9	56.0
UMCC-DLSI	64.7	–
Multi-Objective	**72.8**	66.0
IMS	65.3	67.3
MFS	63.2	65.8

Table 6.3. F-Measure percentage performance on the SemEval-2013 and SemEval-2007 (noun instances) English all-words WSD datatets using WordNet as sense inventory.

SemEval-2007 dataset. We remark that NASARI is an unsupervised system based on global contexts, while IMS is a supervised system based on local contexts. This combination of local and global contexts has already shown to be beneficial for WSD tasks (Hoffart et al., 2011; Pilehvar and Navigli, 2014b; Weissenborn et al., 2015a).

6.2.4 Discussion: global and local contexts

Our method is based on global contexts (we use the whole text as context of the target word to disambiguate), hence it sometimes fails to capture the correct meaning of the word in the cases where the local context appears to be the key to the disambiguation, especially in a fine-grained disambiguation scheme. For instance, in the following sentence taken from the SemEval 2013 Word Sense Disambiguation test set, we find an example where a fine-grained distinction of the target word *behaviour* leads to a mistake by our method which could be solved by exploiting the local context by a supervised system:

(1) The expulsion presumably forged by two players of Real Madrid (Xabi Alonso and Sergio Ramos) in the game played on the 23rd of November against Ajax in European Champions League has caused rivers of ink to be written about if such *behaviour* is or is not unsportmanlike and if, both players should be sanctioned by UEFA.

Our system is not confident enough and hesitates between the sense behaviour^3_n (*The aggregate of the responses or reactions or movements made by an organism in any situation*) and behaviour^4_n (*Manner of acting or controlling yourself*), selecting the latter by a narrow margin. In this case, combining our method with one exploiting local contexts such as IMS would lead to the correct answer.

On the other hand, there are cases where a local-based approach may fail due to the lack of a more global text understanding. We appreciate this phenomenon in the following sentence, also taken from the SemEval 2013 dataset:

(2) This way, and since Real Madrid will finish as leader of its group, both players will fulfil the prescribed *sanction* during the next game of league.

In this case, IMS considers as its highest confidence sense sanction^1_n (*Formal and explicit approval*), which is also the most frequent sense for the noun *sanction*. It gets misled by the closest context and would need to get the higher picture (global context) to fix the error. In this case, Nasari correctly captures the semantics within the text and chooses sanction^2_n (*A mechanism of social control for enforcing a society's standards*).

In both cases the combination of Nasari and IMS gets to the correct answer and in general the combination of both methods shows a consistent improvement over the single system components. In fact, the results of the combination of a knowledge-based global-context disambiguation system (i.e., Nasari) with a state-of-the-art supervised local-context approach (i.e., IMS) proves to be quite robust across datasets, outperforming many strong baselines as we can see from Table 6.3. A more extended analysis of the performance and divergences between knowledge-based and supervised systems can be found in Raganato et al. (2017).

6.3 Multilingual Word Sense Disambiguation Exploiting Comparable or Parallel Corpora

In both WSD and EL tasks, supervised approaches tend to obtain the best performances over standard benchmarks but, as explained in the previous section, from a practical standpoint they lose ground to knowledge-based approaches, which scale better in terms of scope and number of languages. In fact, the development of supervised disambiguation systems depends crucially on the availability of reliable sense-annotated corpora, which are indispensable in order to provide solid training and testing grounds (Pilehvar and Navigli, 2014b). However, hand-labeled sense annotations are notoriously difficult to obtain on a large scale, and manually curated corpora (Miller et al., 1993; Passonneau et al., 2012) have a limited size. Given that scaling the manual annotation process becomes practically unfeasible when both lexicographic and encyclopedic knowledge is addressed (Schubert, 2006), recent years have witnessed efforts to produce larger sense-annotated corpora automatically (Moro et al., 2014; Taghipour and Ng, 2015a; Raganato et al., 2016). Even though these automatic approaches produce noisier corpora, it has been shown that training on them leads to better supervised and semi-supervised models (Taghipour and Ng, 2015b; Yuan et al., 2016; Raganato et al., 2017), as well as to effective embedded representations for senses (Iacobacci et al., 2015; Flekova and Gurevych, 2016). A convenient way of generating sense annotations is to exploit parallel corpora and word alignments (Taghipour and Ng, 2015a): indeed, parallel corpora exist in many flavours (Tiedemann, 2012) and are widely used across the NLP community for a variety of different tasks.

The fundamental idea of work is to exploit comparable or parallel corpora for a high-quality disambiguation. To this end, we tackle two different problems by proposing a single disambiguation model: (1) the disambiguation of definitions coming from different resources and languages, exploiting their cross-lingual and cross-resource complementarities, and (2) the disambiguation of the Europarl parallel corpus[7] (Koehn, 2005), exploiting its translations in different languages.

[7] http://opus.lingfil.uu.se/Europarl.php

Our goal is to obtain a high-quality sense-annotated corpus of both textual definitions and Europarl, constructed using a single multilingual disambiguation model. While language- and resource-specific techniques can certainly be used for disambiguation, they would not be scalable for our goal: the number of models required would add up to the order of hundreds, and there would also be the need for large amounts of sense-annotated data for each language and resource, leading to the so-called knowledge acquisition bottleneck (Gale et al., 1992). Instead, we propose a knowledge-based disambiguation system that exploits at best cross-language cues from any input multilingual text. Our methodology exploits a graph-based disambiguation system along with a refinement based on distributional similarity. This refinement makes use of the multilingual nature of NASARI, which allows its direct integration in multiple languages.

6.3.1 Methodology

In the following we describe our methodology for disambiguating any given target corpus by exploiting any comparable or parallel corpus. Our fully automatic disambiguation pipeline couples a graph-based multilingual joint WSD/EL system, Babelfy (Moro et al., 2014), and a language-independent vector representation of concepts and entities, NASARI (see Section 4). It comprises two stages: multilingual disambiguation and refinement based on distributional similarity.

Stage 1: Multilingual Disambiguation

As a preprocessing step, we part-of-speech tag and lemmatize the whole corpus using TreeTagger (Schmid, 1995).[8] We perform disambiguation at the sentence level. However, instead of disambiguating each sentence in isolation, language by language, we first identify all available translations of a given sentence and then gather these together into a single multilingual text.[9] Then, we disambiguate this multilingual text using Babelfy. Given that Babelfy is capable of handling text with multiple languages at the same time, this multilingual extension effectively increases the amount of context for each sentence, and directly helps in dealing with highly ambiguous words in any particular language (as the translations of these words may be less ambiguous in some different language). Moreover, given the multilingual nature of our sense inventory, Babelfy's high-coherence approach favors naturally sense assignments that are consistent across languages at the sentence level (i.e. those having fewer distinct senses shared by more translations of the same sentence). As a result, we obtain a full, high-coverage version where each disambiguated word or multi-word expression (*disambiguated instance*) is associated with a coherence score, which is computed as the normalized number of connections of a given concept within the sentence.

[8] We rely on the internal preprocessing pipeline of Babelfy for those languages not supported by TreeTagger.

[9] For the disambiguation of definitions, the definitions of the same concept are also considered (see Section 6.3.2).

Stage 2: Similarity-based Refinement

In this stage we aim at improving the sense annotations obtained in the previous step (Section 6.3.1), with a procedure specifically targeted at correcting and extending these sense annotations. In general, graph-based WSD systems, such as Babelfy, have been shown to be heavily biased towards the Most Common Sense (MCS) (Calvo and Gelbukh, 2015). In order to get a handle on this bias and improve our pipeline's disambiguation accuracy we adopt a refinement based on distributional similarity, which is not affected by the MCS. To this end, we exploit the 300-dimensional embedded representations of concepts and entities of NASARI to discard or refine disambiguated instances that are less semantically coherent. For each sentence, we first identify a subset D of high-confidence disambiguations[10] from among those given by Babelfy in the previous step. Then, we calculate the centroid of all the NASARI vectors corresponding to the elements of D, and we re-disambiguate the mentions associated with the remaining low-confidence disambiguated instances (i.e. those not in D), by picking, for each mention w, the concept or entity $\hat{s}$ whose NASARI vector[11] is closest to the centroid of the sentence:

$$\hat{s} = \underset{s \,\in\, S_w}{\operatorname{argmax}}\, cos\left(\frac{\sum_{d \,\in\, D} \vec{d}}{|D|}, \vec{s}\right) \tag{6.2}$$

where S_w is the set of all candidate senses for mention w according to BabelNet. Cosine similarity (cos) is used as similarity measure. Finally, in order to discard less confident annotations, we consider the cosine value associated with each refined disambiguation as confidence score, and use it to compare each disambiguated instance against an empirically validated threshold of 0.75.

As a result, we obtain the refined high-precision version of the sense-annotated corpus where each disambiguated instance is associated with both a coherence score and a distributional similarity score.

6.3.2 SenseDefs: Multilingual corpus of sense-annotated definitions

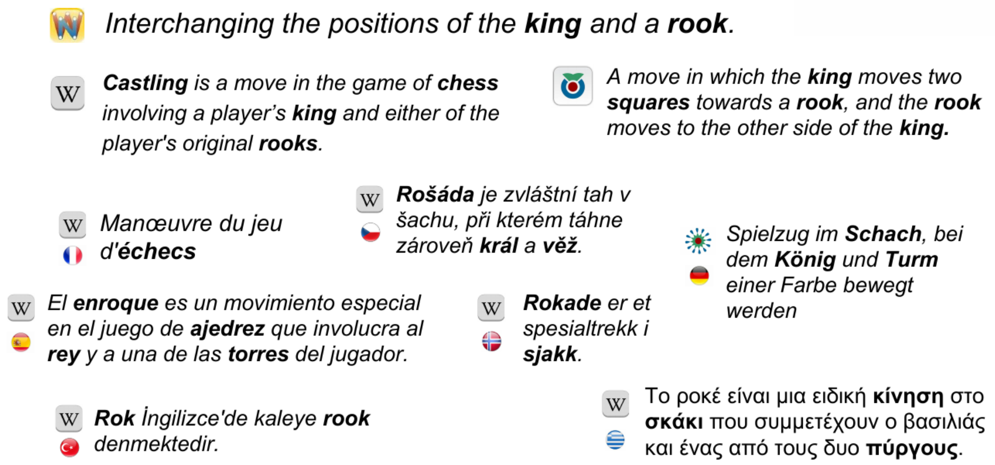
In this section we describe SENSEDEFS, a sense-annotated corpus of textual definitions constructed by applying the methodology described in the previous section. In addition to definition translations, we augmented the context by including definitions from different resources of BabelNet referring to the same concept of entity. This way we leveraged the inter-resource and inter-language mappings provided by BabelNet to combine multiple definitions (drawn from different resources and in different languages) of the same concept or entity; in this way, a much richer context can be associated with each target definition, enabling a high-quality disambiguation.

As an example, consider the following definition of *castling* in chess as provided by WordNet:

Interchanging the positions of the king and a rook. (6.3)

[10] We follow Camacho-Collados et al. (2016a) and consider disambiguated instances with a coherence score above 0.125.

[11] Given a concept or entity s we indicate with $\vec{s}$ its corresponding NASARI vector.



*Interchanging the positions of the **king** and a **rook**.*

***Castling** is a move in the game of **chess** involving a player's **king** and either of the player's original **rooks**.*

*A move in which the **king** moves two **squares** towards a **rook**, and the **rook** moves to the other side of the **king.***

*Manœuvre du jeu d'**échecs***

***Rošáda** je zvláštní tah v šachu, při kterém táhne zároveň **král** a **věž**.*

*Spielzug im **Schach**, bei dem **König** und **Turm** einer Farbe bewegt werden*

*El **enroque** es un movimiento especial en el juego de **ajedrez** que involucra al **rey** y a una de las **torres** del jugador.*

***Rokade** er et spesialtrekk i **sjakk**.*

*Rok İngilizce'de kaleye **rook** denmektedir.*

Το ροκέ είναι μια ειδική **κίνηση** στο **σκάκι** που συμμετέχουν ο βασιλιάς και ένας από τους δυο **πύργους**.

Figure 6.1. Some of the definitions, drawn from different resources and languages, associated with the concept of *castling* in chess through our context enrichment procedure.

The context in this example is limited and it might not be obvious for an automatic disambiguation system that the concept being defined relates to *chess*: for instance, an alternative definition of *castling* where the game of *chess* is explicitly mentioned would definitely help the disambiguation process. Following this idea, given a BabelNet synset, we carry out a *context enrichment* procedure by collecting all the definitions of this synset in every available language and resource, and gathering them together into a single multilingual text. Figure 6.1 gives a pictorial representation of this harvesting process for the concept of *castling* introduced in Example 6.3.

By applying the methodology described in Section 6.3.1 on the whole set of textual definitions in BabelNet for all the available languages, we obtain SenseDefs, a large multilingual corpus of disambiguated glosses. We release two versions of the resource:

- ***Full.*** This high-coverage version provides sense annotations for all content words as provided by Babelfy after the context-rich disambiguation (see Section 6.3.1) and *before* the refinement step.

- ***Refined.*** The refined, high-precision version of SenseDefs, instead, *only* includes the most confident sense annotations as computed by the refinement step (see Section 6.3.1).

Table 6.4 shows some general statistics of the *full* and *refined* versions of SenseDefs, divided by resource. The output of the *full* version is a corpus of 38,820,114 disambiguated glosses, corresponding to 8,665,300 BabelNet synsets and covering 263 languages and 5 different resources (Wiktionary, WordNet, Wikidata, Wikipedia and OmegaWiki). It includes 249,544,708 sense annotations (6.4 annotations per definition on average). The refined version of the resource includes fewer, but more reliable, sense annotations (see Section 6.3.2), and a slightly reduced

	# Glosses		# Annotations	
	Full	**Refined**	**Full**	**Refined**
Wikipedia	29 792 245	28 904 602	223 802 767	143 927 150
Wikidata	8 484 267	8 002 375	22 769 436	17 504 023
Wiktionary	281 756	187 755	1 384 127	693 597
OmegaWiki	115 828	106 994	744 496	415 631
WordNet	146 018	133 089	843 882	488 730
Total	**38 820 114**	**37 334 815**	**249 544 708**	**163 029 131**

Table 6.4. Number of definitions and annotations of the *full* and *refined* versions of SenseDefs.

number of glosses containing at least one sense annotation. Wikipedia is the resource with by far the largest number of definitions and sense annotations, including almost 30 million definitions and over 140 million sense annotations in both versions of the corpus. Additionally, Wikipedia also features textual definitions for the largest number of languages (over 200).

Statistics by language. Figures 6.2 and 6.3 display the number of definitions and sense annotations, respectively, divided by language[12]. As expected, English provides the largest number of glosses and annotations (5.8M glosses and 37.9M sense annotations in the refined version), followed by German and French. Even though the majority of sense annotations overall concern resource-rich languages (i.e. those featuring the largest amounts of definitional knowledge), the language rankings in Figures 6.2 and 6.3 do not coincide exactly: this suggests, on the one hand, that some languages (such as Vietnamese and Spanish, both with higher positions in Figure 6.3 compared to Figure 6.2) actually benefit from a cross-lingual disambiguation strategy; on the other hand, it also suggests that there is still room for improvement, especially for some other languages (such as Swedish or Russian) where the tendency is reversed and the number of annotations is lower compared to the amount of definitional knowledge available.

Table 6.5 shows the number of annotations divided by part-of-speech tag and disambiguation source. In particular, the full version obtained as output of Step 2 comprises two disambiguation sources: Babelfy and the MCS back-off (used for low-confidence annotations). The refined version, instead, removes the MCS back-off, either by discarding or correcting the annotation with NASARI. Additionally, 17% of the sense annotations obtained by Babelfy without resorting to the MCS back-off are also corrected or discarded. Assuming the coverage of the full version to be 100%,[13] the coverage of our system after the refinement step is estimated to be 65.3%. As shown in Table 6.5, discarded annotations mostly consist of verbs, adjectives and

[12] Only the top 15 languages are displayed in the figures.

[13] There is no straightforward way to estimate the coverage of a disambiguation system automatically. In our first step using Babelfy, we provide disambiguated instances for all content words (including multi-word expressions) from BabelNet and also for overlapping mentions. Therefore, the output of our first step, even if it is not perfectly accurate, may be considered to have full coverage.

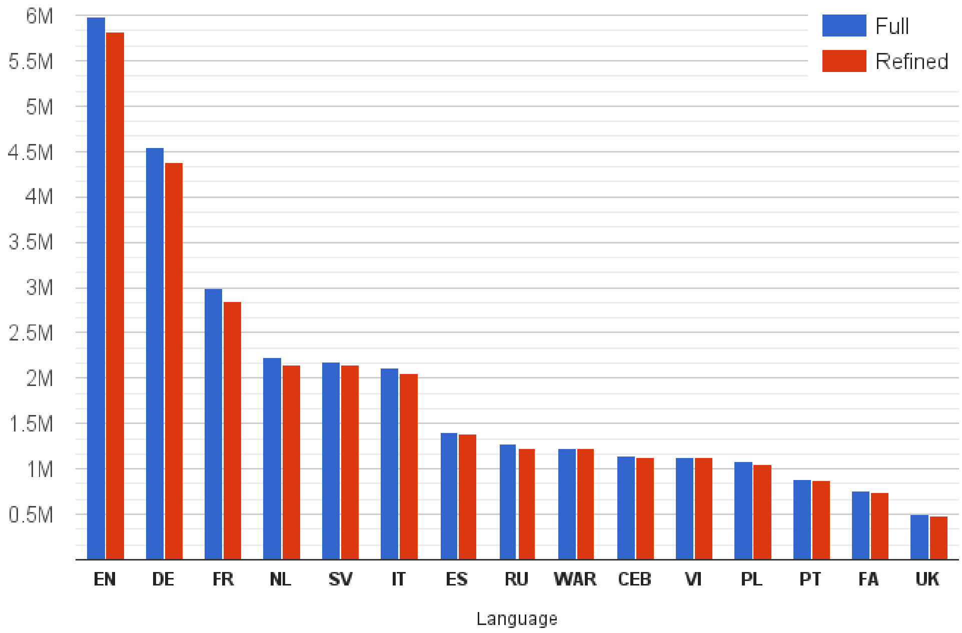


Figure 6.2. Number of definitions by language (top 15 languages).

		All	Nouns	Verbs	Adjectives	Adverbs
Full	**Babelfy**	174 256 335	158 310 414	4 368 488	10 646 921	930 512
	MCS	75 288 373	56 231 910	8 344 930	9 256 497	1 455 036
	Total	**249 544 708**	**214 542 324**	**12 713 418**	**19 903 418**	**2 385 548**
Refined	**Babelfy**	144 637 032	140 111 921	1 326 947	3 064 416	133 748
	NASARI	18 392 099	18 392 099	-	-	-
	Total	**163 029 131**	**158 504 020**	**1 326 947**	**3 064 416**	**133 748**

Table 6.5. Number of annotations by part-of-speech tag (*columns*) and by source (*rows*) before and after refinement.

adverbs, which are often harder to disambiguate as they are very frequently not directly related to the definiendum. In fact, the coverage figure on noun instances is estimated to be 73.9% after refinement.

Intrinsic Evaluation

As intrinsic evaluation we carried out a thorough manual assessment of sense annotation quality in SenseDefs. In our previous study (Camacho-Collados et al., 2016a), we performed a manual evaluation for three languages (English, Italian and Spanish) employing three human judges. Each language was evaluated on a sample of 100 definitions, considering the input of a baseline (i.e. disambiguating definitions in isolation with Babelfy) and our *Full* and *Refined* versions of SenseDefs. In the three languages the context-rich disambiguation achieved better results than the baseline. More importantly, the refinement based on distributional similarity proved

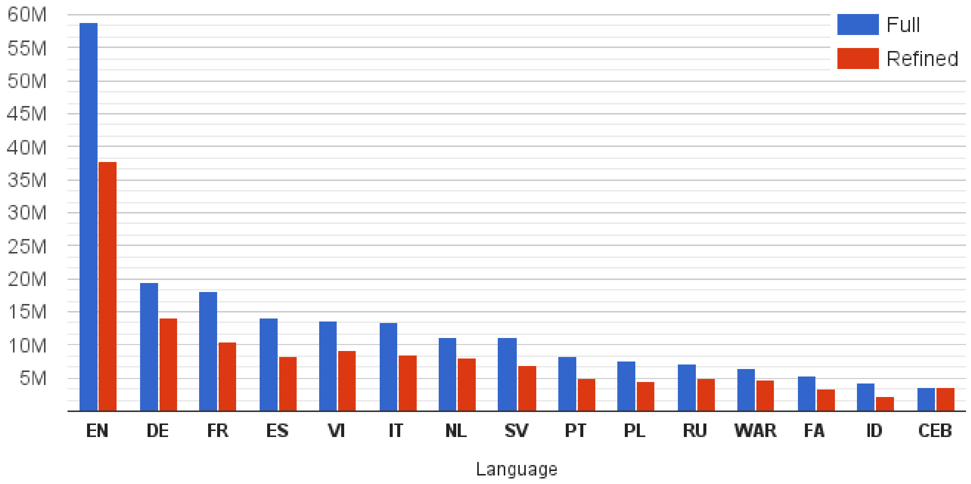


Figure 6.3. Number of annotations by language (top 15 languages).

highly reliable, obtaining a precision over 90% on the three languages, without drastically decreasing the coverage. For this work we have extended that intrinsic evaluation by performing two additional experiments. In the first experiment we extended the manual evaluation of Camacho-Collados et al. (2016a) by increasing the number of definitions, languages and annotators. In the second experiment we performed a large-scale automatic evaluation where we compared our annotations against the manual disambiguation of WordNet glosses.

Manual Evaluation. We carried out an extensive evaluation of sense annotation quality in SenseDefs on four different languages: English, French, Italian and Spanish. To this end, we first randomly sampled 120 definitions for each language. Then, two annotators validated the sense annotations given by SenseDefs (both *Full* and *Refined*) and Babelfy. In contrast to the intrinsic evaluation of Camacho-Collados et al. (2016a), in this case we excluded those annotations coming from the MCS back-off, in order to assess the output explicitly provided by our disambiguation pipeline.

For each item in the sample, each annotator was shown the textual definition, the BabelNet entry for the definiendum, and every non-MCS sense annotation paired with the corresponding BabelNet entry. The annotator had to decide independently, for each sense annotation, whether it was correct (score of 1), or incorrect (score of 0). The disambiguation source (i.e. whether the annotation came from Babelfy in isolation, context-rich disambiguation or NASARI) was not shown. In some special cases where a certain sense annotation was acceptable but a more suitable synset was available, a score of 0.5 was allowed. One recurrent example of these indecisive annotations occurred on multi-word expressions: being designed as a high-coverage all-word disambiguation strategy, Babelfy can output disambiguation decisions over overlapping mentions when confronted with fragments of text having more than one acceptable disambiguation. For instance, the multi-word expression

		#Ann.	Prec.	Rec.*	F1	IAA	
						ROA	κ
EN	Babelfy	671	**84.3**	69.6	76.1	94.6	71.7
	Full	714	80.0	70.2	74.8	94.2	70.1
	Refined	**745**	83.1	**76.1**	**79.5**	95.3	71.9
ES	Babelfy	678	85.8	59.3	70.2	91.4	51.1
	Full	**737**	82.6	62.1	70.9	92.4	66.2
	Refined	725	**86.6**	**64.0**	**73.6**	95.1	63.3
FR	Babelfy	516	84.3	49.8	62.6	97.2	85.7
	Full	568	81.3	52.8	64.0	96.7	86.4
	Refined	**579**	**87.1**	**57.7**	**69.4**	95.1	65.8
IT	Babelfy	540	**81.7**	53.5	64.7	94.5	74.3
	Full	609	73.9	54.5	62.8	92.4	78.0
	Refined	**618**	77.5	**58.1**	**66.4**	94.7	83.0

Table 6.6. Quality of the annotations of SenseDefs for English, Spanish, French and Italian. Recall (*) was computed assuming each content word in a sentence should be associated with a distinct sense. Inter-annotator agreement (IAA) was computed in terms of Relative Observed Agreement (ROA) and Cohen's kappa (κ).

"Commission of the European Union" can be interpreted both as a single mention, referring to the specific BabelNet entity `European Commission`^1_n (executive body of the European Union), and as two mentions, one (*"Commission"*) referring to the BabelNet entry `Parliamentary committee`^1_n (a subordinate deliberative assembly), and the other (*"European Union"*) referring to the the BabelNet entry `European Union`^1_n (the international organization of European countries). In all cases where one part of a certain multi-word expression was tagged with an acceptable meaning, but a more accurate annotation would have been the one associated with the whole multi-word expression, we allowed annotators to assign a score of 0.5 to valid annotations of nested mentions and a score of 1 only to the complete and correct multi-word annotation. Another controversial example of indecision is connected to semantic shifts due to Wikipedia redirections, which cause semantic annotations that are lexically acceptable but wrong from the point of view of semantic roles. For instance, the term *painter* inside Wikipedia redirects to the Wikipedia entry for `Painting` (*Graphic art consisting of an artistic composition made by applying paints to a surface*), while the term *Basketball player* redirects to the Wikipedia entry for `Basketball` (*Sport played by two teams of five players on a rectangular court*). These redirections are also exploited by Babelfy as acceptable disambiguation decisions (a policy that is often used in Entity Linking, especially in Wikipedia-specific settings) and, as such, they are also allowed a score of 0.5.

Once the annotations were completed, we calculated the Inter Annotator Agreement (IAA) between the two annotators of each language by means of Relative Observed Agreement (ROA), calculated as the proportion of equal answers, and Cohen's kappa (Cohen, 1968, κ). Finally, the two annotators in each language adjudicated the answers which were judged with opposite values. Table 6.6 shows the results of this manual evaluation. In the four languages, our refined version of the corpus achieved the best overall results, consistently with the results of the previous intrinsic evaluation (Camacho-Collados et al., 2016a). SenseDefs achieved over 80% precision in three of the four considered languages, both in its full and refined versions. For Italian the precision dropped to 73.9% and 77.5%, respectively, probably due to its lower coverage in BabelNet. Finally, it is worth noting that, for all the examined languages, both the full and refined versions of SenseDefs provided more annotations than using the Babelfy baseline on isolated definitions.

Automatic evaluation: WordNet glosses. To complement the manual intrinsic evaluation, we performed an additional large-scale automatic evaluation. We compared the WordNet annotations given by SenseDefs[14] with the manually-crafted annotations of the disambiguated glosses from the Princeton Gloss Corpus[15]. Similarly to the previous manual evaluation, we included a baseline based on Babelfy disambiguating the definitions sentence-wise in isolation and using the pre-trained models[16] of the IMS (Zhong and Ng, 2010, It Makes Sense) supervised disambiguation system. IMS uses a SVM classifier including features based on surrounding words and local collocations. As in our previous experiment, we did not considered the annotations for which the MCS back-off strategy was activated on any of the comparison systems. Finally, as baseline we include the results of WordNet first sense (i.e. MCS) for the annotations disambiguated by each system. The MCS baseline has been shown to be hard to beat, especially for knowledge-based systems (Raganato et al., 2017). However, this baseline, which is computed from a sense-annotated corpus, is only available for the English WordNet. Therefore, it is not possible to use this MCS baseline accurately for languages other than English, and resources other than WordNet for which sense-annotated data is not available or is very scarce.

Table 6.7 shows the accuracy results (computed as the number of annotations corresponding to the manual annotations divided by the total number of overlapping annotations) of SenseDefs, Babelfy and IMS on the Princeton Gloss Corpus. SenseDefs achieved an accuracy of 76.4%, both in its full and refined versions. Nevertheless, the refined version attained a larger coverage, disambiguating a larger amount of instances. This result is relatively high considering the nature of the corpus, consisting of short and concise definitions for which the context is clearly limited. In fact, even if not directly comparable, the best systems in standard WSD SemEval competitions (where full documents are given as context to disambiguate) tend to obtain considerably less accurate results (Edmonds and Cotton, 2001; Snyder and Palmer, 2004; Pradhan et al., 2007; Navigli et al., 2013; Moro and Navigli, 2015). In fact, even though results are not directly comparable[17], IMS achieved an accuracy which is considerably lower than our system's performance and also lower compared to its performance on standard benchmarks (Raganato et al., 2017).

[14] As explained in Section 6.3.1, our disambiguation pipeline annotates with BabelNet synsets, hence its coverage is larger than only WordNet. This implies that some annotations are not comparable to those inside the WordNet glosses.

[15] `http://wordnet.princeton.edu/glosstag.shtml`

[16] Downloaded from `http://www.comp.nus.edu.sg/~nlp/corpora.html#onemilwsd`. We used the models from the One Million Sense-Tagged Instances as training corpus.

[17] Recall that our system annotates with BabelNet synsets and hence the set of disambiguation candidates is larger than IMS and the MCS baseline. This also makes the set of annotations differ with respect to IMS.

	#WN Annotations	Accuracy	MCS-Acc.
SenseDefs$_{\textbf{Full}}$	162 819	**76.4**	66.1
SenseDefs$_{\textbf{Refined}}$	169 696	**76.4**	65.2
Babelfy	130 236	69.1	65.6
IMS	275 893	56.1	55.2

Table 6.7. Accuracy and number of compared WordNet annotations on the Princeton Gloss Corpus. On the right the accuracy of MCS and IMS on the same sample.

This result highlights the added difficulty of disambiguating definitions, as they do not provide enough context for an accurate disambiguation in isolation. Only our disambiguation pipeline, which does not make use of any sense-annotated data, proves reliable in this experiment, comfortably outperforming the MCS baseline on the same annotations.

Extrinsic Evaluation

We also evaluated extrinsically the effectiveness of SenseDefs (both the full and refined versions of the resource) by making use of its sense annotations within two Natural Language Processing tasks: Open Information Extraction and Sense Clustering.

Open Information Extraction. In this experiment we investigated the impact of our disambiguation approach on the definitional corpus used as input for the pipeline of DefIE (Delli Bovi et al., 2015b). The original OIE pipeline of the system takes as input an unstructured corpus of textual definitions, which are then preprocessed one-by-one to extract syntactic dependencies and disambiguate word senses and entity mentions. After this preprocessing stage, the algorithm constructs a syntactic-semantic graph representation for each definition, from which subject-verb-object triples (relation instances) are eventually extracted. As highlighted in the previous section, poor context of particularly short definitions may introduce disambiguation errors in the preprocessing stage, which then tend to propagate and reflect on the extraction of both relations and relation instances. To assess the quality of our disambiguation strategy as compared to the standard approach, we modified the implementation of DefIE to consider our disambiguated instances instead of executing the original disambiguation step, and then we evaluated the results obtained at the end of the pipeline in terms of quality of relation and relation instances.

We first selected a random sample of 150 textual definitions from our disambiguated corpus. We generated a baseline for the experiment by discarding all disambiguated instances from the sample, and treating the sample itself as an unstructured text of textual definitions which we used as input for DefIE, letting the original pipeline of the system carry out the disambiguation step. Then we carried out the same procedure using, instead, the modified implementation for which our disambiguated instances are taken into account. In both cases, we ran the extraction

	# Glosses	# Triples	# Relations
DefIE + glosses	**150**	**340**	**184**
DefIE	146	318	171

Table 6.8. Extractions of DefIE on the evaluation sample.

	Relation	Relation Instances
DefIE + glosses	**0.872**	**0.780**
DefIE	0.865	0.770

Table 6.9. Precision of DefIE on the evaluation sample.

algorithm of DefIE and evaluated the output in terms of both relations and relation instances. Following Delli Bovi et al. (2015b), we employed two human judges and performed the same evaluation procedure described therein over the set of distinct relations extracted from the sample, as well as the set of extracted relation instances.

Results reported in Tables 6.8 and 6.9 show a slight but consistent improvement resulting from our disambiguated glosses over both the number of extracted relations and triples and over the number of glosses with at least one extraction (Table 6.8), as well as over the estimated precision of such extractions (Table 6.9). Context-rich disambiguation of glosses across resources and languages enabled the extraction of 6.5% additional instances from the sample (2.26 extractions on the average from each definition) and, at the same time, increased the estimated precision of relation and relation instances over the sample by ∼1%.

Sense Clustering. The second experiment, instead, evaluated the refined version of SenseDefs on the sense clustering task. For this experiment we used the semantic representations of NASARI (see Section 4). In particular, we reconstructed the vectorial representations of NASARI by, 1) enriching the semantic network used in the original implementation with the refined sense annotations of SenseDefs, and 2) running again the NASARI pipeline to generate the vectors. We then evaluated these on the sense clustering task (see Section 7.1), improving over the original NASARI vectors and achieving state-of-the-art results.

6.3.3 EuroSense: Europarl sense-annotated corpus

For building EuroSense we use the methodology described in Section 6.3.1 for disambiguating the Europarl (Koehn, 2005) parallel corpus. Europarl, which is one of the most popular multilingual corpora, was originally designed to provide aligned parallel text for Machine Translation (MT) systems. Extracted from the proceedings of the European Parliament, the latest release of the Europarl corpus comprises parallel text for 21 European languages, with more than 743 million tokens overall. Apart from its prominent role in MT as a training set, the Europarl corpus has been used for cross-lingual WSD (Lefever and Hoste, 2010, 2013), including, more recently, preposition sense disambiguation (Gonen and Goldberg, 2016), and widely exploited to develop cross-lingual word embeddings (Hermann and Blunsom, 2014; Gouws et al., 2015; Coulmance et al., 2015; Vyas and Carpuat, 2016; Vulić and Korhonen, 2016; Artetxe et al., 2016) as well as multi-sense embeddings (Ettinger et al., 2016; Šuster et al., 2016). Our aim is to augment Europarl with sense-level information for multiple languages, thereby constructing a large-scale sense-annotated multilingual corpus which has the potential to boost both WSD and MT research. Unlike previous cross-lingual approaches, we do not rely on word alignments against a pivot language, but instead leverage all languages at the same time in a joint disambiguation procedure that is subsequently refined using distributional similarity. As a result of our disambiguation pipeline we obtain and make available to the community EuroSense, a multilingual sense-annotated corpus with almost 123 million sense annotations of more than 155 thousand distinct concepts and named entities drawn from the multilingual sense inventory of BabelNet, and covering all the 21 languages of the Europarl corpus. As such EuroSense constitutes, to our knowledge, the largest corpus of its kind.

Table 6.10 reports general statistics on EuroSense regarding both its full and refined versions[18]. Joint multilingual disambiguation with Babelfy generated more than 215M sense annotations of 247k distinct concepts and entities, while similarity-based refinement retained almost 123M high-confidence instances (56.96% of the total), covering almost 156k distinct concepts and entities. 42.40% of these retained annotations were corrected or validated using distributional similarity. As expected, the distribution over parts of speech is skewed towards nominal senses (64.79% before refinement and 81.79% after refinement) followed by verbs (19.26% and 12.22%), adjectives (11.46% and 5.24%) and adverbs (4.48% and 0.73%). We note that the average coherence score increases from 0.19 to 0.29 after refinement, suggesting that distributional similarity tends to favor sense annotations that are also consistent across different languages. Table 6.10 also includes language-specific statistics on the 4 languages of the intrinsic evaluation, where the average lexical ambiguity ranges from 1.12 senses per lemma (German) to 2.26 (English) and, as expected, decreases consistently after refinement.

Interestingly enough, if we consider all the 21 languages, the total number of distinct lemmas covered is more than twice the total number of distinct senses: this is a direct consequence of having a unified, language-independent sense inventory (BabelNet), a feature that sets EuroSense apart from previous multilingual sense-annotated corpora Otegi et al. (2016). Finally we note from the global figures on the number of covered senses that 109 591 senses (44.2% of the total) are not covered by the English sense annotations: this suggests that EuroSense relies heavily on multilinguality in integrating concepts or named entities that are tied to specific social or cultural aspects of a given language (and hence would be underrepresented in an English-specific sense inventory).

We assessed the quality of EuroSense's sense annotations both intrinsically, by means of a manual evaluation on four samples of randomly extracted sentences in different languages, as well as extrinsically, by augmenting the training set of a state-of-the-art supervised WSD system Zhong and Ng (2010) and showing that it leads to consistent performance improvements over two standard WSD benchmarks.

[18] These two versions correspond to the same versions of SenseDefs, corresponding to the two steps of the disambiguation (see Section 6.3.1).

		Total	EN	FR	DE	ES
Full	# Annotations	215 877 109	26 455 574	22 214 996	16 888 108	21 486 532
	Distinct lemmas covered	567 378	60 853	30 474	66 762	43 892
	Distinct senses covered	247 706	138 115	65 301	75 008	74 214
	Average coherence score	0.19	0.19	0.18	0.18	0.18
Refined	# Annotations	122 963 111	15 441 667	12 955 469	9 165 112	12 193 260
	Distinct lemmas covered	453 063	42 947	23 603	50 681	31 980
	Distinct senses covered	155 904	86 881	49 189	52 425	52 859
	Average coherence score	0.29	0.28	0.25	0.28	0.27

Table 6.10. General statistics on EuroSense before *(full)* and after refinement *(refined)* for all the 21 languages. Language-specific figures are also reported for the 4 languages of the intrinsic evaluation.

	English		**French**		**German**		**Spanish**	
	Prec.	**Cov.**	**Prec.**	**Cov.**	**Prec.**	**Cov.**	**Prec.**	**Cov.**
Babelfy	76.1	100	59.1	100	80.4	100	67.5	100
EuroSense (full)	80.3	100	67.9	100	84.6	100	76.7	100
EuroSense (refined)	**81.5**	75.0	**71.8**	63.5	**89.3**	53.8	**82.5**	62.9

Table 6.11. Precision *(Prec.)* and coverage *(Cov.)* of EuroSense, manually evaluated on a random sample in 4 languages. Precision is averaged between the two judges, and coverage is computed assuming each content word in the sense inventory to be a valid disambiguation target.

Intrinsic Evaluation

In order to assess annotation quality directly, we carried out a manual evaluation on 4 different languages (English, French, German and Spanish) with 2 human judges per language. We sampled 50 random sentences across the subset of sentences in EuroSense featuring a translation in all 4 languages, totaling 200 sentences overall. For each sentence, we evaluated all sense annotations both before and after the refinement stage, along with the sense annotations obtained by a baseline that disambiguates each sentence in isolation with Babelfy. Overall, we manually verified a total of 5818 sense annotations across the three configurations (1518 in English, 1564 in French, 1093 in German and 1643 in Spanish). In every language the two judges agreed in more than 85% of the cases, with an inter-annotator agreement in terms of Cohen's kappa (Cohen, 1960) above 60% in all evaluations (67.7% on average). Results, reported in Table 6.11, show that joint multilingual disambiguation improves consistently over the baseline. The similarity-based refinement boosts precision even further, at the expense of a reduced coverage (whereas both Babelfy and the baseline attempt an answer for every disambiguation target). Over the 4 languages, sense annotations appear to be most reliable for German, which is consistent with its lower lexical ambiguity on the corpus (see Section 6.10).

Extrinsic Evaluation: Word Sense Disambiguation

We additionally carried out an extrinsic evaluation of EuroSense by using its refined sense annotations for English as a training set for a supervised all-words

	SemEval-2013	**SemEval-2015**
$\text{IMS}_{\text{SemCor}}$	65.3	69.3
$\text{IMS}_{\text{OMSTI}}$	65.0	69.1
$\text{IMS}_{\text{EuroSense}}$	**66.4**	**69.5**
UKB	62.9	63.3
MCS	63.0	67.8

Table 6.12. F-Score on all-words WSD.

WSD system, It Makes Sense (Zhong and Ng, 2010, IMS). Following Taghipour and Ng (2015a), we started with SemCor (Miller et al., 1993) as initial training dataset, and then performed a subsampling of EuroSense up to 500 additional training examples per word sense. We then trained IMS on this augmented training set and tested on the two most recent standard benchmarks for all-words WSD: the SemEval-2013 task 12 (Navigli et al., 2013) and the SemEval-2015 task 13 (Moro and Navigli, 2015) test sets. As baselines we considered IMS trained on SemCor only and OMSTI, the sense-annotated dataset constructed by Taghipour and Ng (2015a) which also includes SemCor. Finally, we report the results of UKB, a knowledge-based system (Agirre et al., 2014).[19] As shown in Table 6.12, IMS trained on our augmented training set consistently outperforms all baseline models, showing the reliability of EuroSense as training corpus, even against sense annotations obtained semi-automatically (Taghipour and Ng, 2015a).

6.4 Conclusion

In this section we presented a pipeline for exploiting comparable and parallel corpora for a high-quality multilingual disambiguation. The method is based on two steps, combining graph-based and distributional cues. Nasari (Section 4) is used on the second step based on distributional similarity, reinforcing the use of these vector representations of concepts and entities on a multilingual setting. By exploiting this pipeline we constructed two sense-annotated corpora: SenseDefs and EuroSense.

SenseDefs (Section 6.3.2) is a large-scale multilingual corpus of disambiguated textual definitions (or glosses). We obtained high-quality sense annotations by exploiting our pipeline designed to exploit cross-resource and cross-language complementarities of multiple textual definitions associated with a given definiendum. By leveraging the structure of a wide-coverage semantic network and sense inventory like BabelNet, we obtained a corpus of textual definitions coming from multiple sources and multiple languages, fully disambiguated with BabelNet synsets. SenseDefs, to the best of our knowledge, is the largest available corpus of its kind. Moreover, the choice of BabelNet as sense inventory not only provides wide-coverage sense annotations of both a lexicographic and encyclopedic nature. Indeed, since BabelNet is a merger of various different resources, including WordNet and Wikipedia, these annotations are also expandable to any of these resources and can be easily converted

[19] We include its two implementations using the full WordNet graph and the disambiguated glosses of WordNet as connections: default and word by word (*w2w*).

via BabelNet's inter-resource mappings. We evaluated SenseDefs extensively, with both intrinsic (manual and automatic) and extrinsic experiments (on OIE and sense clustering tasks).

Finaly, we presented EuroSense (Section 6.3.3), a large multilingual sense-annotated corpus based on Europarl, and constructed automatically via our multilingual disambiguation pipeline. Crucially, EuroSense relies on the wide-coverage unified sense inventory of BabelNet, which enabled the disambiguation process to exploit at best parallel text and enforces cross-language coherence among sense annotations. We evaluated EuroSense both intrinsically and extrinsically, showing that it provides reliable sense annotations that improve supervised models for WSD.

Chapter 7

Knowledge Base Enrichment

In this chapter we explore the use of knowledge-based representations for improving and enriching current knowledge bases. Creating and adding information into knowledge bases is a labour-intensive task. Therefore, the inclusion of automatic methods for this work becomes essential. For this task sense, concept and entity representations can play a decisive role, as they act as a bridge between NLP and knowledge resources. In particular, for this chapter we study the problems of sense clustering (Section 7.1), domain labeling (Section 7.2), hypernym discovery (Section 7.3) and collocation discovery (Section 7.4).

7.1 Sense Clustering

Some lexical resources suffer from a high granularity (i.e. high degree of polisemy) of their sense inventories. For example, in WordNet there exist a sense of *street* including *sidewalks* and another one without *sidewalks*. This high granularity could possibly affect the performance of applications relying on their sense inventories (Palmer et al., 2007; Hovy et al., 2013). In fact, the use of a reduced and coarser sense inventory has been shown to be beneficial in various applications (Flekova and Gurevych, 2016; Pilehvar et al., 2017)[1]. Nevertheless, obtaining an optimal granularity of sense inventories remains an open problem. This optimal granularity may vary from application to application, thus tailoring the sense inventory to a specific application is often required, e.g. Cocos et al. (2017a) for lexical substitution. To deal with this problem we propose a simple method based on semantic similarity which makes use of sense representations.

7.1.1 Background

A large amount of work has been dedicated to reduce the granularity of WordNet. Earlier works approached this problem by exploiting WordNet-specific similarity and relatedness techniques (Mihalcea and Moldovan, 2001; Agirre and Lopez, 2003; McCarthy, 2006). Navigli (2006) leveraged textual definitions (or glosses) for clustering senses of the Oxford Dictionary of English. In order to combine the best of previous approaches, Snow et al. (2007) proposed a complex method which made

[1] See also Chapter 8.

use of a wide variety of WordNet-based and corpus-based features integrated into a Support Vector Machine (SVM) classifier.

Following this line of work, Dandala et al. (2013a) proposed another SVM classifier using a similar set of features, combined with Wikipedia-specific features, for clustering Wikipedia pages. Among the Wikipedia-based features, they explicitly exploited the multilinguality of Wikipedia to propose multilingual features which proved very effective for the task. As in the earlier attempts on this area, Pilehvar et al. (2013) proposed a simple use of semantic similarity between semantic representations of synsets for reducing the granularity of WordNet's sense inventory. In particular, we propose a similar method to make use of sense vector representations and semantic similarity for clustering the Wikipedia sense inventory. Our method is flexible enough as it can cluster senses on both an unsupervised fashion and also supervised which may lead to a more precise granularity depending on the application, without making explicit use of resource-specific tools or techniques.

7.1.2 Methodology

As mentioned in the previous section, we exploit sense vector representations and semantic similarity for sense clustering. Given the nature of both SW2V and Nasari, we could seamlessly perform sense clustering in BabelNet, WordNet or Wikipedia. Following earlier works Pilehvar et al. (2013); Dandala et al. (2013a), we view sense clustering as a binary classification task in which given a pair of senses the task is to decide if they have to be merged or not. In the usual setting of clustering, where senses which are semantically related are clustered together, we rely on our similarity scale and simply cluster a pair of items (synsets, senses or pages) together provided that their similarity exceeds the middle point in our similarity scale, i.e., 0.5 in the scale of [0, 1], and with a minimum overlap between vectors of five dimensions. In specific sense clustering settings, this middle-point threshold may be changed to another value, or determined using a tuning dataset.

7.1.3 Experimental setting

For the evaluation we focus on Wikipedia. Given the granularity of the Wikipedia sense inventory, clustering related senses may improve systems which take Wikipedia as their knowledge source (Hovy et al., 2013). Wikipedia-based Word Sense Disambiguation (Mihalcea, 2007; Dandala et al., 2013b) is an example of an application which may benefit from this sense inventory clustering.

Wikipedia can be considered as a sense inventory wherein the different meanings of a word are denoted by the articles listed in its disambiguation page (Mihalcea and Csomai, 2007). Starting from these Wikipedia disambiguation pages and with the help of human annotation, Dandala et al. (2013a) created two Wikipedia sense clustering datasets. In these datasets, clustering is viewed as a binary classification task in which all possible pairings of senses of a word are annotated whether they should be clustered or not[2]. The first dataset, which we will refer to as **500-pair**

[2] McCarthy et al. (2016) showed how on some cases word senses are not easily clusterable. In our work we do not study the partitionability of word senses and consider all word senses of a lemma to be clusterable as defined in the gold standard.

dataset, contains 500 pairs, 357 of which are set to belong to the same cluster or *clustered*, and the remaining 143 to *not clustered.* The second dataset, referred to as the **SemEval** dataset, is based on a set of highly ambiguous words taken from SemEval evaluations Mihalcea (2007) and consists of 925 pairs, 162 of which are positively labeled (clustered). *Parameter_(computer_programming)-Parameter* and *Fatigue(medical)-Fatigue(safety)* are two sample pairs of Wikipedia pages that should be merged.

As explained above, our systems are based on semantic similarity (see Section 4.3) for the sense clustering task. Two senses (in this case two Wikipedia pages) are set to be clustered if their similarity is greater than a threshold γ, which is set by default to the middle point of the similarity scale (i.e., 0.5). As mentioned earlier, some applications may required a more precise clusterization of a given sense inventory. This could be achieved using both SW2V (see Chapter 5) and NASARI$_{\text{embed}}$ synset embeddings on a supervised setting. In order to explore the complementarity of SW2V and NASARI, we additionally test both representations together (NASARI$_{\text{embed}}$+SW2V). To this end, given a Wikipedia page, we concatenate the corresponding SW2V and NASARI vectors, resulting on a 600-dimensional vector.[3]

In order to set the optimal value of γ, we follow Dandala et al. (2013a) and use the first 500-pairs sense clustering dataset for tuning. We set the threshold γ to 0.35 for SW2V and NASARI$_{\text{embed}}$+SW2V and 0.7 for NASARI$_{\text{embed}}$, respectively, which are the values leading to the highest F-Measure among all values from 0 to 1 with a 0.05 step size on the 500-pair dataset. Likewise, we set a 0.3 threshold for the SensEmbed (Iacobacci et al., 2015) comparison system.[4]

7.1.4 Results

Our experiments are carried out on the 500-pair and SemEval datasets. We set two naive baselines: one considering all the pairs as positive or clustered (**Baseline$_{\text{cluster}}$**), and another one doing the opposite, i.e., not clustering any of the test pairs (**Baseline$_{\text{no-cluster}}$**). In addition to SensEmbed, we also compare our system to two systems proposed by Dandala et al. (2013a). Both systems exploit the structure and content of the Wikipedia pages by using a multi-feature Support Vector Machine classifier trained on an automatically-labeled dataset. This first system is totally monolingual (it only makes use of English Wikipedia pages), while the second system also exploits Wikipedia multilinguality[5]. We will refer to the first system as **SVM-monolingual** and to the second system as **SVM-multilingual**. Finally, we also report the results of NASARI lexical vectors enriched with SENSEDEFS (see Section 6.3.2). NASARI uses Wikipedia ingoing links and the BabelNet taxonomy in the process of obtaining contextual information for a given concept (see Section 4.2). We simply enriched the BabelNet taxonomy with the refined version of the

[3] If a Wikipedia page is not covered by one of the systems, we simply include the null vector instead.

[4] SensEmbed consists of BabelNet sense embeddings downloaded from `http://lcl.uniroma1.it/sensembed/`. See Section 5.4 for more details on SensEmbed.

[5] For this second system we report their results for the system configuration which exploits Wikipedia pages in four different languages (English, German, Spanish, and Italian).

Measure	System type	500-pair		SemEval	
		Acc.	**F1**	**Acc.**	**F1**
NASARI	unsupervised	83.8	70.5	87.4	63.1
NASARI$_{\text{lexical}}$	unsupervised	81.6	65.4	85.7	57.4
NASARI$_{\text{unified}}$	unsupervised	82.6	69.5	87.2	63.1
NASARI$_{\text{embed}}$	unsupervised	81.2	65.9	86.3	45.5
NASARI+SENSEDEFS	unsupervised	**86.0**	**74.8**	**88.1**	64.7
Baseline$_{\text{no-cluster}}$	-	71.4	0.0	82.5	0.0
Baseline$_{\text{cluster}}$	-	28.6	44.5	17.5	29.8
NASARI$_{\text{embed}}$	supervised	-	-	87.0	62.5
SW2V	supervised	-	-	87.8	63.9
SensEmbed	supervised	-	-	82.7	40.3
NASARI$_{\text{embed}}$+SW2V	supervised	-	-	88.0	**65.2**
SVM-monolingual	supervised	77.4	-	83.5	-
SVM-multilingual	supervised	84.4	-	85.5	-

Table 7.1. Accuracy (Acc.) and F-Measure (F1) percentages of different systems on the two manually-annotated English Wikipedia sense clustering datasets.

disambiguated glosses in English. These disambiguated glosses contain synsets that are highly semantically connected with the definiendum, which makes them particularly suitable for enriching a semantic network. The rest of the pipeline for obtaining lexical semantic representations (i.e. lexical specificity applied to the contextual information) remained unchanged.[6]

Table 7.1 shows the results obtained for the Wikipedia sense clustering task in the 500-pair and SemEval datasets. The results are shown in terms of accuracy (number of correctly labeled pairs divided by total number of instance pairs) and F-Measure (harmonic mean of precision and recall). As we can see from the Table, NASARI in its unsupervised setting achieves a very high accuracy, outperforming both systems of Dandala et al. (2013a) on the SemEval dataset and SVM-monolingual on the 500-pair dataset. Only the supervised system of Dandala et al. (2013a) using information of Wikipedia pages in different languages outperforms our main combined NASARI system in terms of accuracy (no F-Measure results were reported) by a narrow margin. NASARI, in any of the three variants, comfortably outperforms the naive baselines in terms of both accuracy and F-Measure. When comparing our three systems, the combination of both lexical and unified vectors outperforms both single-handed components. However, both lexical- and unified- based systems (and embedding-based) also prove to be highly competitive single-handed, outperforming all baselines on the SemEval dataset, including the multilingual approach of Dandala et al. (2013a). Interestingly, the enrichment produced by SENSEDEFS proved highly beneficial,

[6] As an additional advantage, by integrating the high-precision disambiguated glosses into the NASARI pipeline we obtained a new set of vector representations for BabelNet synsets, increasing its initial coverage (4.4M synsets covered by the original NASARI, compared to 4.6M synsets covered by NASARI enriched with our disambiguated glosses).

significantly improving on the original results obtained by NASARI. Moreover, NASARI+SenseDefs obtained the best performance overall, outperforming the SVM-based systems of Dandala et al. (2013a) in terms of accuracy in both datasets, and all the supervised systems on the SemEval dataset.

As for the supervised sense embeddings comparison, SW2V outperforms all comparison systems according to both measures, including Nasari$_{\text{embed}}$ by a small margin and the sense representations of SensEmbed using the same setup, corpus and underlying lexical resource. This confirms the capability of both SW2V and Nasari to accurately capture the semantics of word senses on this sense-specific task. More interestingly, the concatenation of both SW2V and Nasari$_{\text{embed}}$ proves highly effective, achieving the best results overall, even outperforming NASARI+SenseDefs in terms of F-Measure. This result highlights the complementarity of both SW2V and Nasari, as the signals used for their learning process capture complementarity information. While Nasari extracts and process knowledge direcly from the lexical resources, SW2V extracts knowledge directly from text corpora.

7.2 Domain Labeling

Since the early days of Natural Language Processing (NLP) and Machine Learning, generalizing a given algorithm or technique has been extremely challenging. One of the main factors that has led to this issue in NLP has been the wide variety of domains for which data are available (Jiang and Zhai, 2007). Algorithms trained on the business domain are not to be expected to work well in biology, for example. Moreover, even if we manage to obtain a balanced training set across domains, our algorithm may not be as effective on some specific domain as if it had been trained on that same target domain. This issue has become even more challenging and significant with the rise of supervised learning techniques. These techniques are fed with large amounts of data and ought to be able generalize to various target domains. Several studies have proposed regularization frameworks for domain adaptation in NLP (Daumé III and Marcu, 2006; Daumé III, 2007; Lu et al., 2016). In this work we tackle this problem but approach it from a different angle. Our main goal is to integrate domain information into lexical resources, which, in turn, could enable a semantic clusterization of training data by domain, a procedure known as multi-source domain adaptation (Crammer et al., 2008). In fact, adapting algorithms to a particular domain has already proved essential in standard NLP tasks such as Word Sense Disambiguation (Magnini et al., 2002; Agirre et al., 2009b; Faralli and Navigli, 2012), Text Categorization (Navigli et al., 2011), Sentiment Analysis (Glorot et al., 2011; Hamilton et al., 2016), or Hypernym Discovery (Espinosa-Anke et al., 2016a), *inter alia.*

7.2.1 Background

Enriching lexical resources with domain knowledge has been studied in previous works. For example, the domain annotation of WordNet has already been carried out in previous studies (Magnini and Cavaglià, 2000; Bentivogli et al., 2004; Tufiş

Animals	Language and linguistics
Art, architecture and archaeology	Law and crime
Biology	Literature and theatre
Business, economics, and finance	Mathematics
Chemistry and mineralogy	Media
Computing	Meteorology
Culture and society	Music
Education	Numismatics and currencies
Engineering and technology	Philosophy and psychology
Food and drink	Physics and astronomy
Games and video games	Politics and government
Geography and places	Religion, mysticism and mythology
Geology and geophysics	Royalty and nobility
Health and medicine	Sport and recreation
Heraldry, honors, and vexillology	Transport and travel
History	Warfare and defense

Table 7.2. The set of thirty-two domains.

et al., 2008). Domain information is also available in IATE[7], a European Union inter-institutional terminology database. The domain labels of IATE are based on the Eurovoc thesaurus[8] and were introduced manually. The fact that each of these approaches involves manual curation/intervention limits their extension to other resources, and therefore to downstream applications.

We, instead, have developed an automatic hybrid distributional and graph-based method for encoding domain information into lexical resources. In this work we aim at annotating BabelNet, a large unified lexical resource which integrates WordNet and other resources[9] such as Wikipedia and Wiktionary, augmenting the initial coverage of WordNet by two orders of magnitude.

7.2.2 Methodology

Our goal is to enrich lexical resources with domain information. To this end, we rely on BabelNet 3.0, which merges both encyclopedic (e.g. Wikipedia) and lexicographic resources (e.g. WordNet). The main unit in BabelNet, similarly to WordNet, is the synset, which is a set of synonymous words corresponding to the same meaning (e.g., {*midday, noon, noontide*}). In contrast to WordNet, a BabelNet synset may contain lexicalizations coming from different resources and languages. Therefore, the annotation of a BabelNet synset could directly be expanded to all its associated resources.

As domains of knowledge, we opted for domains from the *Wikipedia featured*

[7] `http://iate.europa.eu/`

[8] `http://eurovoc.europa.eu/drupal/?q=navigation&cl=en`

[9] See Section 2.3 for an overview of BabelNet and its integrated resources.

articles page[10]. This page contains a set of thirty-two domains of knowledge.[11] Table 7.2 shows the set of thirty-two domains. For each domain, there is a set of Wikipedia pages associated (127 on average). For instance, the Wikipedia pages *Kolkata* and *Oklahoma* belong to the `Geography` domain[12]. Our methodology for annotating BabelNet synsets with domains is divided into two steps: (1) we apply a distributional approach to obtain an extensive distribution of domain labels in BabelNet (Section 7.2.2), and (2) we complement this first step with a set of heuristics to improve the coverage and correctness of the domain annotations (Section 7.2.2).

Distributional similarity

We exploit NASARI$_{\text{lexical}}$ vectors for this prior step. In order to obtain a full distribution for each BabelNet synset, i.e. a list of ranked domains associated, each domain is first associated with a given vector. Then, the Wikipedia pages from the featured articles page are leveraged as follows. First, all Wikipedia pages associated with a given domain are concatenated into a single text. Second, a lexical vector is constructed for each text by applying lexical specificity over the bag-of-word representation of the text, as explained in Section 4.1.2. Finally, given a BabelNet synset s, the similarity between its respective NASARI lexical vector and the lexical vector of each domain is calculated using the Weighted Overlap comparison measure (see Section 4.1.5).[13]

This enables us to obtain, for each BabelNet synset, scores for each domain label denoting their importance. For notational brevity, we will refer to the domain whose similarity score is highest across all domains as its *top domain*. For instance, the top domain of the BabelNet synset corresponding to *rifle* is `Warfare`, while its second domain is `Engineering`. In order to increase precision, initially we only tag those BabelNet synsets whose maximum score is higher than 0.35.[14]

Heuristics

We additionally propose three heterogeneous heuristics to improve the quality and coverage of domain annotations. These heuristics are applied in cascade (in the same order as they appear on the text) over the labels provided on the previous step.

1. **Taxonomy.** This first heuristic is based on the BabelNet hypernymy structure, which is an integration of various taxonomies: WikiData, WordNet and MultiWiBi (Flati et al., 2016). The main intuition is that, in general, synsets connected by a hypernymy relation tend to share the same domain (Magnini and Cavaglià, 2000).[15] This taxonomy-based heuristic is intended to both

[10] `https://en.wikipedia.org/wiki/Wikipedia:Featured_articles`

[11] Biography domains are not considered.

[12] For simplicity we refer to each domain with its first word (e.g., `Geography` to refer to *Geography and Places*).

[13] Weighted Overlap has been proved to suit interpretable vectors better than cosine (Pilehvar and Navigli, 2015; Camacho-Collados et al., 2015b).

[14] This value was set through observation to increase precision but without drastically decreasing recall.

[15] In WordNet this property is satisfied most of the times. However, in Wikipedia, especially given its large amount of entities, this is not always the case. For instance, *Microsoft* is a *company*

increase coverage and refine the quality of synsets annotated by the distributional approach. First, if all the hypernyms (at least two) of a given synset share the same top domain, this synset is annotated (or re-annotated) with that domain. Second, if the top domain of an annotated synset is different from at least two of its hypernyms, this domain tag is removed.

2. **Labels.** Some Wikipedia page titles include general information about the page between parentheses. This text between parentheses is known as a label. For example, the Wikipedia page *Orange (telecommunications)* has *telecommunications* as its label. In BabelNet these labels are kept in the main senses of many synsets, information which is valuable for deciding their domain. For those synsets sharing the same label, we create a distribution of domains, i.e. each label is associated with its corresponding synsets and their domains. Then, we tag (or retag) all the synsets containing the given label provided that the most frequent domain for that label gets a number of instances higher than 80% of the total of instances containing the same label.[16] As an example, before applying this heuristic the label *album* contained 14192 synsets which were pre-tagged with a given domain. From those 14192 synsets, 14166 were pre-tagged with the *Music* domain (99.8%). Therefore, the remaining 26 synsets and all the rest containing the *album* label were tagged or re-tagged with the *Music* domain.

3. **Propagation.** In this last step we propagate the domain annotations over the BabelNet semantic network. First, given an unannotated input synset, we gather a set with all its neighbours in the BabelNet semantic network. Then we retrieve the domain with the highest number of synsets associated among all annotated synsets in the set. Similarly to the previous heuristic, if the number of synsets of such domain amounts to 80% of the whole set, we tag the input synset with that domain. Otherwise, we repeat the process with the second-level neighbours and, if still not found, with its third-level neighbours.

7.2.3 BabelDomains: Statistics

We applied the methodology described in Section 7.2.2 on BabelNet 3.0. This led to a total of 2.68M synsets tagged with a domain. Note that this number greatly improves on the number given in previous studies for WordNet. In our approach, in addition to WordNet, we provide annotations for other lexical resources such as Wikipedia or Wiktionary. Table 7.3 shows some statistics of the synsets tagged in each step of the whole domain annotation process. The largest number of annotated synsets were obtained in the first distributional step (1.31M) and the final propagation (1.11M), while the taxonomy and labels heuristics contributed to not only increasing the coverage, but also to refining potentially dubious annotations.

(tagged with the `Business` domain) but it would arguably better have `Computing` as its top domain.

[16] This threshold is set in order to improve the precision of the system, as there are labels which might be ambiguous within a domain (e.g., country names).

	New	Re-ann.	Removed
Distributional	1.31M	-	-
Taxonomy	164K	32K	7K
Labels	94K	4K	-
Propagation	1.11M	-	-
Total	2.68M	-	-

Table 7.3. Number of tagged synsets (*new*, *re-annotated* and *removed*) in each of the domain annotation steps.

7.2.4 Intrinsic evaluation

In this section we evaluate the quality of BabelDomains intrinsically on two different lexical resources: BabelNet and WordNet. In Section 7.3 we further use and evaluate these domain annotations on the hypernym discovery task.

Evaluation datasets

In order to evaluate the performance of our domain labeling approach we constructed two gold standard domain labeled datasets:

- **WordNet domain-labeled dataset** For the construction of this dataset, we took the WordNet 3.0 synsets which were manually tagged with domains. The domain set of WordNet differs from our set of domains (see Table 7.2 for our final domain set). Therefore, we performed a manual mapping from the WordNet domains to our domain set in order to make them comparable. Domains in WordNet were mapped to one of our domains provided that the surface form of the WordNet domain matched the surface form of one of our domain labels. For instance, a WordNet synset whose domain was either *Business*, *Economics* or *Finance* was to be mapped to the domain *Business, economics, and finance*. There are WordNet synsets tagged with more than one domain in WordNet, but we considered only those with a single domain in WordNet for the gold standard construction. As a result, we obtained a gold standard dataset of 1540 WordNet synsets tagged with our domain set[17].

- **BabelNet domain-labeled dataset** In order to have a more realistic distribution of BabelNet synsets comprising not only synsets which belong to the WordNet sense inventory, we created a second gold-standard dataset based on BabelNet. For this, we randomly sampled 200 BabelNet synsets with at least one English lexicalization from the set of all 6.5M possible BabelNet synsets. Of these, 65% were integrated in Wikipedia and only 1.5% belonged to WordNet (the remaining synsets were mostly integrated in WikiData only). Two annotators manually labeled these 200 synsets. They were instructed to mark each synset with a single domain only. Any disagreements were adjudicated in a final phase by the two annotators. The inter-annotator agreement

[17] There is no overlap between these 1540 WordNet synsets and the Wikipedia seeds taken by our system.

was computed to be 86%, which may be viewed as an upper-bound for the performance of automatic systems.

Comparison systems

As comparison systems we included a baseline based on Wikipedia (Wikipedia-idf). This baseline first constructs a *tf-idf*-weighted bag-of-word vector representation of Wikipedia pages and, similarly to our distributional approach, calculates its similarity with the concatenation of all Wikipedia pages associated with a domain in the Wikipedia featured articles page.[18] We additionally compared with WN-Domains-3.2 (Magnini and Cavaglià, 2000; Bentivogli et al., 2004), which is the latest released version of WordNet Domains[19]. However, this approach involves manual curation, both in the selection of seeds and correction of errors. In order to enable a fair comparison, we report the results of a system based on its main automatic component. This baseline takes annotated synsets as input and propagates them through the WordNet taxonomy (WN-TaxoProp). Likewise, we report the results of the same baseline by propagating through the BabelNet taxonomy (BN-TaxoProp). These two systems were evaluated by 10-fold cross validation on the corresponding datasets. Finally, we include the results of the distributional approach performed in the first step of our methodology (Section 7.2.2).

Results

Table 7.4 shows the results of our system and four comparison systems. The initial distributional step based on NASARI provides the grounds for a high-quality domain annotations, with precision figures around 80% in both datasets. Overall, BabelDomains achieves the best F-Measure results by increasing the coverage, with precision figures above 80% on both WordNet and BabelNet datasets. These results improve the results achieved by applying the first step of distributional similarity only, highlighting that the inclusion of the heuristics was beneficial. These precision figures are especially relevant considering the large set of domains (32) used in our methodology. By analyzing the errors, we realized that our system tends to provide domains close to the gold standard. For instance, the synset referring to *entitlement*[20] was tagged with the `Business` domain instead of the gold `Law`. Other domains which produced imperfect choices due to their close proximity were `Mathematics-Computing` and `Animals-Biology`. As regards the generally low recall on the BabelNet dataset, we found that it was mainly due to the nature of the dataset, including many isolated synsets which are hardly used in practice.

7.2.5 Conclusion

In this section we presented BabelDomains, a resource that provides unified domain information in lexical resources. Our method exploits at best the knowledge available in these resources by combining distributional and graph-based approaches. We evaluated the accuracy of our approach on two resources, BabelNet and WordNet.

[18] For the annotation of WordNet we used the direct Wikipedia-WordNet mapping from BabelNet.

[19] `http://wndomains.fbk.eu/`

[20] Defined as *right granted by law or contract (especially a right to benefits).*

	WordNet			BabelNet		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
BabelDomains	81.7	68.7	**74.6**	**85.1**	32.0	**46.5**
Distributional	**84.0**	59.8	69.9	78.1	16.0	26.6
Wikipedia-idf	45.9	29.7	36.1	8.8	6.5	7.5
WN-TaxoProp	71.3	70.7	71.0	-	-	-
BN-TaxoProp	73.5	**73.5**	73.5	48.3	**37.2**	42.0
WN-Domains-3.2	93.6	64.4	76.3	-	-	-

Table 7.4. Precision, Recall and F-Measure percentages of different systems on the gold standard WordNet and BabelNet domain-labeled datasets.

The results showed that our unified resource provides reliable annotations, improving over various competitive baselines. In the future we plan to extend our set of domains with more fine-grained information, providing a hierarchical structure following the line of Bentivogli et al. (2004).

As an extrinsic evaluation we used BabelDomains to cluster training data by domain prior to applying a supervised hypernym discovery system. This pre-clustering proved crucial for finding accurate hypernyms in a distributional vector space. We are planning to further use our resource for multi-source domain adaptation on other NLP supervised tasks. Additionally, since BabelNet and most of its underlying resources are multilingual, we plan to use our resource in languages other than English.

7.3 Hypernym Discovery

Lexical taxonomies (taxonomies henceforth) are graph-like hierarchical structures where terms are nodes, and are typically organized over a predefined merging or splitting criterion (Hwang et al., 2012). By embedding cues about how we perceive concepts, and how these concepts generalize in a domain of knowledge, these resources bear a capacity for generalization that lies at the core of human cognition (Yu et al., 2015) and have become key in Natural Language Processing (NLP) tasks where inference and reasoning have proved to be essential. In fact, taxonomies have enabled a remarkable number of novel NLP techniques, e.g. the contribution of WordNet to lexical semantics (Pilehvar et al., 2013; Yu and Dredze, 2014) as well as various tasks, from word sense disambiguation (Agirre et al., 2014) to information retrieval (Varelas et al., 2005), question answering (Harabagiu et al., 2003) and textual entailment (Glickman et al., 2005). To date, the application of taxonomies in NLP has consisted mainly of, on one hand, formally representing a domain of knowledge (e.g. `Food`), and, on the other hand, constituting the semantic backbone of large-scale knowledge repositories such as ontologies or Knowledge Bases (KBs).

In domain knowledge formalization, prominent work has made use of the web (Kozareva and Hovy, 2010), lexico-syntactic patterns (Navigli and Velardi, 2010), syntactic evidence (Luu Anh et al., 2014), graph-based algorithms (Fountain and Lapata, 2012; Velardi et al., 2013; Bansal et al., 2014) or popularity of web sources

(Luu Anh et al., 2015). As for enabling large-scale knowledge repositories, this task often tackles the additional problem of disambiguating word senses and entity mentions. Notable approaches of this kind include Yago (Suchanek et al., 2007b), WikiTaxonomy (Ponzetto and Strube, 2008), and the Wikipedia Bitaxonomy (Flati et al., 2014). In addition, while not being taxonomy learning systems *per se*, semi-supervised systems for Information Extraction such as Nell (Carlson et al., 2010) rely crucially on taxonomized concepts and their relations within their learning process.

Taxonomy learning is roughly based on a two-step process, namely *is-a* (hypernymic) *relation detection*, and *graph induction*. The hypernym detection phase has gathered much interest not only for taxonomy learning but also for lexical semantics. It has been addressed by means of pattern-based methods[21] (Hearst, 1992; Snow et al., 2005; Kozareva and Hovy, 2010; Carlson et al., 2010; Boella and Di Caro, 2013; Espinosa-Anke et al., 2016c), clustering (Yang and Callan, 2009) and graph-based approaches (Fountain and Lapata, 2012; Velardi et al., 2013). Moreover, work stemming from distributional semantics introduced notions of linguistic regularities found in vector representations such as word embeddings (Mikolov et al., 2013a). In this area, supervised approaches, arguably the most popular nowadays, learn a feature vector between term-hypernym vector pairs and train classifiers to predict hypernymic relations. These pairs may be represented either as a concatenation of both vectors (Baroni et al., 2012), difference (Roller et al., 2014), dot-product (Mikolov et al., 2013c), or including additional linguistic information for LSTM-based learning (Shwartz et al., 2016).

In this work we propose TaxoEmbed[22], a hypernym detection algorithm based on sense embeddings, which can be easily applied to the construction of lexical taxonomies. It is designed to discover hypernymic relations by exploiting linear transformations in embedding spaces (Mikolov et al., 2013b) and, unlike previous approaches, leverages this intuition to learn a specific *semantically-aware transformation matrix* for each domain of knowledge. Our best configuration (ranking first in two thirds of the experiments conducted) considers two training sources: (1) Manually curated pairs from Wikidata (Vrandečić and Krötzsch, 2014); and (2) Hypernymy relations from a KB which integrates several Open Information Extraction (OIE) systems (Delli Bovi et al., 2015a). Since our method uses a very large semantic network as reference sense inventory, we are able to perform jointly hypernym extraction and disambiguation, from which expanding existing ontologies becomes a trivial task. Compared to word-level taxonomy learning, TaxoEmbed results in more refined and unambiguous hypernymic relations at the sense level, with a direct application in tasks such as semantic search. Evaluation (both manual and automatic) shows that we can effectively replicate the Wikidata *is-a* branch, and capture previously unseen relations in other reference taxonomies (Yago or WiBi).

[21] The terminology is not entirely unified in this respect. In addition to *pattern-based* (Fountain and Lapata, 2012; Bansal et al., 2014; Yu et al., 2015), other terms like *path-based* (Shwartz et al., 2016) or *rule-based* (Navigli and Velardi, 2010) are also used.

[22] Data and source code available from the following link: `http://wwwusers.di.uniroma1.it/~dellibovi/taxoembed/`.

7.3.1 Background

Pattern-based methods for hypernym identification exploit the joint co-ocurrence of term and hypernym in text corpora. Building up on Hearst's patterns (Hearst, 1992), these approaches have focused on, for instance, exploiting templates for harvesting candidate instances which are ranked via mutual information (Etzioni et al., 2005), training a classifier with WordNet hypernymic relations combined with syntactic dependencies (Snow et al., 2006), or applying a doubly-anchored method (Kozareva and Hovy, 2010), which queries the web with two semantically related terms for collecting domain-specific corpora. Syntactic information is also used for supervised definition and hypernym extraction (Navigli and Velardi, 2010; Boella and Di Caro, 2013), or together with Wikipedia-specific heuristics (Flati et al., 2016). One of the main drawbacks of these methods is that they require both term and hypernym to co-occur in text within a certain window, which strongly hinders their recall. Higher recall can be achieved thanks to distributional methods, as they do not have co-occurrence requirements. In addition, they can be tailored to cover any number of predefined semantic relations such as co-hyponymy or meronymy (Baroni and Lenci, 2011), but also cause-effect or entity-origin (Hendrickx et al., 2009). However, they are often more imprecise and seem to perform best in discovering broader semantic relations (Shwartz et al., 2016).

One way to surmount the issue of generality was proposed by Fu et al. (2014), who explored the possibility to learn a *hypernymic* transformation matrix over a word embeddings space. As shown empirically in Fu et al.'s original work, the hypernymic relation that holds for the pair (*dragonfly*, *insect*) differs from the one of e.g. (*carpenter*, *man*). Prior to training, their system addresses this discrepancy via k-means clustering using a held-out development set for tuning.

The previously described methods for hypernym and taxonomy learning operate inherently at the surface level. This is partly due to the way evaluation is conducted, which is often limited to very specific domains with no integrative potential (e.g. taxonomies in `food`, `science` or `equipment` from Bordea et al. (2015)), or restricted to lists of word pairs. Hence, a drawback of surface-level taxonomy learning, apart from ambiguity issues, is that they require additional and error-prone steps to identify semantic clusters (Fu et al., 2014).

Alternatively, recent advances in OIE based on disambiguation and deeper semantic analysis (Nakashole et al., 2012; Grycner and Weikum, 2014; Delli Bovi et al., 2015b) have shown their potential to construct taxonomized disambiguated resources both at node and at relation level. However, in addition to their inherently broader scope, OIE approaches are designed to achieve high coverage, and hence they tend to produce noisier data compared to taxonomy learning systems.

In our sense-based approach, instead, not only do we leverage an unambiguous vector representation for hypernym discovery, but we also take advantage of a domain-wise clustering strategy to directly obtain specific term-hypernym training pairs, thereby substantially refining this step. Additionally, we exploit the complementary knowledge of OIE systems by incorporating high-confidence relation triples drawn from OIE-derived resources, yielding the best average configuration as evaluated on ten different domains of knowledge.

7.3.2 Preliminaries

TaxoEmbed leverages the vast amounts of training data available from structured and unstructured knowledge resources, along with the mapping among these resources and a state-of-the-art vector representation of word senses.

BabelNet (see Section 2.3) constitutes our sense inventory, as it is currently the largest single multilingual repository of named entities and concepts, integrating various resources such as WordNet, Wikipedia or Wikidata. As in WordNet, BabelNet is structured in synsets. Each synset is composed of a set of words (*lexicalizations* or *senses*) representing the same meaning. For instance, the synset referring to *the members of a business organization* is represented by the set of senses *firm, house, business firm.* BabelNet contains around 14M synsets in total. We exploit BabelNet[23] as (1) A repository for the manually-curated hypernymic relations included in Wikidata; (2) A semantic pivot of the integration of several OIE systems into one single resource, namely Kb-Unify; and (3) A sense inventory for the SensEmbed vector representations. In the following we provide further details about each of these resources.

Training Data

Wikidata. Wikidata (Vrandečić and Krötzsch, 2014) is a document-oriented semantic database operated by the Wikimedia Foundation with the goal of providing a common source of data that can be used by other Wikimedia projects. Our initial training set $\mathcal{W}$ consists of the hypernym branch of Wikidata, specifically the version included in BabelNet. Each term-hypernym $\in \mathcal{W}$ is in fact a pair of BabelNet synsets, e.g. the synset for *Apple* (with the company sense), and the concept *company.*

KB-Unify. KB-Unify[24] (Delli Bovi et al., 2015a, KB-U) is a knowledge-based approach, based on BabelNet, for integrating the output of different OIE systems into a single unified and disambiguated knowledge repository. The unification algorithm takes as input a set **K** of OIE-derived resources, each of which is modeled as a set of ⟨entity, relation, entity⟩ triples, and comprises two subsequent stages: in the first *disambiguation* stage, each KB in **K** is linked to the sense inventory of BabelNet by disambiguating its relation argument pairs; in the following *alignment* stage, equivalent relations across different KB in **K** are merged together. As a result, KB-U generates a KB of triples where arguments are linked to the corresponding BabelNet synsets, and relations are replaced by *relation synsets* of semantically similar OIE-derived relation patterns. The original experimental setup of KB-Unify included Nell (Carlson et al., 2010) as one of its input resources: since Nell features its own manually-built taxonomic structure and relation type inventory (hence its own *is-a* relation type), we identified the relation synset containing Nell's *is-a*[25] and then drew from the unified KB all the corresponding triples, which we denote as $\mathcal{K}$. These triples constitute, similarly as in the previous case, a set of term-hypernym pairs au-

[23] We use BabelNet 3.0 release version in our experiments.

[24] `http://lcl.uniroma1.it/kb-unify`

[25] represented by the relation `generalizations`.

tomatically extracted from OIE-derived resources, with a disambiguation confidence of above 0.9 according to the disambiguation strategy described in the original paper.

Initially, $|\mathcal{W}| = 5{,}301{,}867$ and $|\mathcal{K}| = 1{,}358{,}949$.

Sense vectors

SensEmbed (Iacobacci et al., 2015) constitutes the sense embeddings space that we use for training our hypernym detection algorithm. Vectors in the SensEmbed space, denoted as $\mathcal{S}$, are latent continuous representations of word senses based on the Word2Vec architecture (Mikolov et al., 2013a), which was applied on a disambiguated Wikipedia corpus. Each vector $\vec{v} \in \mathcal{S}$ represents a BabelNet sense, i.e. a synset along with one of its lexicalizations (e.g. *album_chart_bn:00002488n*). While other knowledge-based sense embeddings could have been used, we decided to use SensEmbed as to keep the linguistic regularities from Word2Vec (Mikolov et al., 2013d). This differs from unsupervised approaches (Huang et al., 2012; Tian et al., 2014; Neelakantan et al., 2014) that learn sense representations from text corpora only and are not mapped to any lexical resource, limiting their application in our task.

7.3.3 Methodology

Our approach can be summarized as follows. First, we take advantage of a clustering algorithm for allocating each BabelNet synset of the training set into a domain cluster C (Section 7.3.3). Then, we expand the training set by exploiting the different lexicalizations available for each BabelNet synset (Section 7.3.3). Finally, we learn a cluster-wise linear projection (a *hypernym transformation matrix*) over all pairs (term-hypernym) of the expanded training set (Section 7.3.3).

Domain Clustering

Fu et al. (2014) induced semantic clusters via k-means, where k was tuned on a development set. Instead, we aim at learning a function sensitive to a predefined knowledge domain, under the assumption that vectors clustered with this criterion are likely to exhibit similar semantic properties (e.g. similarity). First, we allocate each synset into its most representative domain, which is achieved by exploiting the set of domains available in BabelDomains (see Table 7.4). In particular, we exploit the domain annotations based on the distributional similarity approach making use of the Nasari vectors (see Section 7.2). As shown in Table 7.3, by following this methodology almost 1.5 million synsets are labelled with a domain.

Training Data Expansion

Prior to training our model, we benefit from the fact that a given BabelNet synset may be associated with a fixed number of lexicalizations or senses, i.e. different ways of referring to the same concept, to expand our set of training pairs. For instance, the synset b associated with the concept *music_album* is represented by the set of lexicalizations $\mathcal{L}_b = \{$album, music_album … album_project$\}$. We take advantage of this synset representation to expand each term-hypernym synset pair. For each term-hypernym pair, both concepts are expanded to their given lexicalizations and thus, each synset pair term-hypernym in the training data is expanded to a set of $|\mathcal{L}_t|.|\mathcal{L}_h|$ sense training pairs.

This expansion step results in much larger sets $\mathcal{W}^*$ and $\mathcal{K}^*$, where $|\mathcal{W}^*| =$ 18,291,330 and $|\mathcal{K}^*| =$ 15,362,268. Specifically, they are 3 and 11 times bigger than the original training sets described in Section 7.3.2. These numbers are higher than those reported in recent approaches for hypernym detection, which exploited Chinese semantic thesauri along with manual validation of hypernym pairs (Fu et al., 2014) (obtaining a total of 1,391 instances), or pairs from knowledge resources such as Wikidata, Yago, WordNet and DBpedia (Shwartz et al., 2016), where the maximum reported split for training data (70%) amounted to 49,475 pairs.

Learning a Hypernym Detection Matrix

The gist of our approach lies on the property of current semantic vector space models to capture relations between vectors, in our case hypernymy. This can be found even in disjoint spaces, where this property has been exploited for machine translation (Mikolov et al., 2013b) or language normalization (Tan et al., 2015). For our purposes, however, instead of learning a global linear transformation function in two spaces over a broad relation like hypernymy, we learn a function sensitive to a given domain of knowledge. Thus, our training data becomes restricted to those term-hypernym BabelNet sense pairs $\left(x^d, y^d\right) \in C_d \times C_d$, where C_d is the cluster of BabelNet synsets labelled with the domain d.

For each domain-wise expanded training set T^d, we construct a hyponym matrix $\mathbf{X}^d = [\vec{x}^d_1 \dots \vec{x}^d_n]$ and a hypernym matrix $\mathbf{Y}^d = [\vec{y}^d_i \dots \vec{y}^d_n]$, which are composed of the corresponding sense vectors of the training pairs $\left(x^d_i, y^d_i\right) \in C_d \times C_d, 0 \leq i \leq n$.

Under the intuition that there exists a matrix Ψ so that $\vec{y}^d = \Psi \vec{x}^d$, we learn a transformation matrix for each domain cluster C_d by minimizing:

$$\min_{\Psi^C} \sum_{i=1}^{|T^d|} \|\Psi^C \vec{x}^d_i - \vec{y}^d_i\|^2 \tag{7.1}$$

Then, for any unseen term x^d, we obtain a ranked list of the most likely hypernyms of its lexicalization vectors $\vec{x_j}^d$, using as measure cosine similarity:

$$\text{argmax}_{\vec{v} \in \mathcal{S}} \frac{\vec{v} \cdot \Psi^C \vec{x_j}^d}{||\vec{v}||||\Psi^C \vec{x_j}^d||} \tag{7.2}$$

At this point, we have associated with each sense vector a ranked list of candidate hypernym vectors. However, in the (frequent) cases in which one synset has more than one lexicalization, we need to condense the results into one single list of candidates, which we achieve with a simple ranking function $\lambda(\cdot)$, which we compute as $\lambda(\vec{v}) = \frac{cos(\vec{v}, \Psi^C \vec{x}^d)}{rank(\vec{v})}$, where $rank(\vec{v})$ is the rank of $\vec{v}$ according to its cosine similarity with $\Psi^C \vec{x}^d$.

The above operations allow us to cast the hypernym detection task as a ranking problem. This is also particularly interesting to enable a flexible evaluation framework where we can combine highly demanding metrics for the quality of the candidate given at a certain rank, as well as other measures which consider the rank of the first valid retrieved candidate.

7.3.4 Automatic evaluation

In this section we assess the ability of TaxoEmbed to return valid hypernyms for a given unseen term using Wikidata as training and test data.

Experimental setting

For each domain, we retain 5k, 10k, 15k, 20k and 25k Wikidata term-hypernym training pairs for different experiments, and evaluate on 250 test pairs for each of the 10 domains. Moreover, we aim at improving TaxoEmbed by including 1k and 25k extra OIE-derived training pairs per domain (generating two more systems, namely 25k+K^d_{1k} and 25k+K^d_{25k}). These OIE-derived instances are those contained in KB-U (see Section 7.3.2). Moreover, in order to quantify the empirically grounded intuition of the need to train a cluster-wise transformation matrix Fu et al. (2014), we also introduce an additional configuration at 25k (25k+K^r_{50k}), where we include 50k additional pairs randomly from KB-U, and two more settings with only random pairs coming from Wikidata (100k$^r_{wd}$) and KB-U (100k+$^r_{kbu}$).

We also include a distributional supervised baseline[26] based on word analogies (Mikolov et al., 2013a), computed as follows. First, we calculate the difference vector of each training sense vector pair ($\vec{x}^d$,$\vec{y}^d$) of a given domain d. Then, we average all the difference vectors of all training pairs to obtain a global vector $\vec{V}_d$ for the domain d. Finally, given a test term t we calculate the closest vector of the sum of the corresponding term vector and $\vec{V}_d$:

$$\hat{h} = \text{argmax}_{\vec{h} \in \mathcal{S}} cos(\vec{V}_d + \vec{t}, h) \tag{7.3}$$

This baseline has shown to capture different semantic relations and to improve as training data increases (Mikolov et al., 2013a).

Evaluation metrics. We computed, for each domain and for the above configurations, the following metrics: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and R-Precision (R-P). These measures provide insights on different aspects of the outcome of the task, e.g. how often valid hypernyms were retrieved in the first positions of the rank (MRR), and if there were more than one valid hypernym, whether this set was correctly retrieved, (MAP and R-P)[27].

Results and discussion

We summarize the main outcome of our experiments in Table 7.5. Results suggest that the performance of TaxoEmbed increases as training data expands. This is consistent with the findings shown in Mikolov et al. (2013b), who showed a substantial improvement in accuracy in the machine translation task by gradually

[26] Using the 25k domain-filtered expanded Wikidata pairs as training set.

[27] See Bian et al. (2008) for an in-depth analysis of these metrics.

	`art`			`biology`			`education`			`geography`			`health`		
Train	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P
5k	0.12	0.12	0.12	0.63	0.63	0.59	0.00	0.00	0.00	0.08	0.07	0.07	0.08	0.08	0.07
15k	0.21	0.20	0.18	**0.84**	0.72	0.79	0.22	0.22	0.21	0.15	0.14	0.14	0.08	0.07	0.07
25k	**0.29**	0.27	**0.26**	**0.84**	**0.83**	**0.81**	0.33	0.32	0.30	**0.23**	**0.22**	**0.21**	0.09	0.09	0.08
25k+K^d_{1k}	**0.29**	**0.28**	**0.26**	**0.84**	0.80	0.79	0.32	0.29	0.27	0.22	**0.22**	**0.21**	0.09	0.09	0.08
25k+K^d_{25k}	0.26	0.24	0.22	0.70	0.63	0.56	**0.38**	**0.36**	**0.33**	0.15	0.13	0.12	0.11	0.11	0.10
25k+K^r_{50k}	0.28	0.26	0.24	0.82	0.77	0.72	0.36	0.33	0.30	0.17	0.16	0.16	**0.12**	0.11	0.10
100k$^r_{wd}$	0.00	0.00	0.00	**0.84**	0.81	0.77	0.00	0.00	0.00	0.01	0.01	0.01	0.07	0.06	0.06
100k$^r_{kbu}$	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	**0.12**	0.12	0.11
Baseline	0.13	0.12	0.10	0.58	0.57	**0.57**	0.10	0.10	0.09	0.12	0.09	0.05	0.07	**0.13**	**0.14**

	`media`			`music`			`physics`			`transport`			`warfare`		
Train	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P
5k	0.28	0.28	0.27	0.10	0.10	0.09	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01
15k	0.14	0.13	0.12	0.08	0.07	0.07	0.36	0.35	0.34	0.25	0.23	0.21	0.01	0.01	0.01
25k	0.46	0.45	0.43	0.30	0.28	0.26	**0.41**	**0.40**	**0.38**	0.46	0.43	0.39	**0.05**	**0.05**	**0.04**
25k+K^d_{1k}	0.43	0.42	0.41	**0.32**	**0.30**	**0.28**	0.39	0.38	0.37	0.47	0.44	0.40	0.04	0.04	0.01
25k+K^d_{25k}	0.52	**0.51**	0.49	0.26	0.25	0.23	0.37	0.36	0.34	0.48	0.45	0.41	0.04	0.03	0.03
25k+K^r_{50k}	0.46	0.45	0.43	0.29	0.28	0.25	0.31	0.30	0.29	**0.52**	**0.49**	**0.46**	**0.05**	0.04	**0.04**
100k$^r_{wd}$	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.01
100k$^r_{kbu}$	0.08	0.07	0.07	0.01	0.01	0.00	0.00	0.00	0.00	0.10	0.10	0.10	0.00	0.00	0.00
Baseline	**0.57**	0.43	**0.52**	0.03	0.03	0.03	0.05	0.04	0.04	0.29	0.25	0.21	0.04	0.04	**0.04**

Table 7.5. Overview of the performance of TaxoEmbed using different training data samples.

increasing the training set. Additionally, the improvement of TaxoEmbed over the baseline is consistent across most evaluation domain clusters and metrics, with domain-filtered data from KB-U contributing to the learning process in about two thirds of the evaluated configurations. These are very encouraging results considering the noisy nature of OIE systems, and that the resource we obtained from KB-U is the result of error-prone steps such as Word Sense Disambiguation and Entity Linking, as well as relation clustering. More importantly, Additionally, the domain clustering based on Nasari vectors[28] proves essential. In fact, training directly without pre-clusterization leads to very poor results, despite being trained on a larger sample: it only provides competitive results in `Biology` only, arguably due to the distribution of Wikidata where biology items are over-represented.

As far as the individual domains are concerned, the `biology` domain seems to be easier to model than the rest, likely due to the fact that fauna and flora are areas where hierarchical division of species is a field of study in itself, which traces back to Aristotelian times (Mayr, 1982), and therefore has been constantly refined over the years. Also, it is notable how well the 100k$^r_{wd}$ configuration performs on this domain. This is the only domain in which training with no semantic awareness gives good results. We argue that this is highly likely due to the fact that a vast amount

[28] In Camacho-Collados and Navigli (2017) we performed additional experiments showing that the results can be further improved by integrating the heuristics presented in Section 7.2.2

of synsets are allocated into the `biology` cluster (60% of them, and up to 80% in hypernym position). This produces the so-called lexical memorization phenomenon (Levy et al., 2015b), as the system memorizes prototypical biology-related hypernyms like *taxon* as valid hypernyms for many concepts. This contrasts with the lower presence of other domains, e.g. 5% in `media`, 4% in `music`, or 2% in `transport`.

Another remarkable case involves the `education` and `media` domains, which experience the highest improvement when training with KB-U (5 and 6 MRR points, respectively). One of the main sources for *is-a* relations in KB-U is Nell, which contains a vast amount of relation triples between North American academic entities (professors, sports teams, alumni, donators; as well as media celebrities). Many of these entities are missing in Wikidata, and relations among them encoded in Nell are likely to be correct because in most cases these are unambiguous entities which occur in the same communicative contexts. For example, leveraging KB-U we were able to include the pair (*university_of_north_wales*, *four_year_college*), which is absent in Wikidata. In fact, many high quality *is-a* pairs like this can be found in KB-U for these two domains.

We also computed P@k (number of valid hypernyms on the first k returned candidates), where k ranges from 1 to 5. Numbers are on the line of the results shown in Table 7.5 and therefore are not provided in detail. The main trend we found is showcased in Figure 7.1, which shows an illustrative example from the `transport` domain. As we can see, all values of k exhibit a similar performance curve, with a gradual increase of performance as the training set becomes larger.

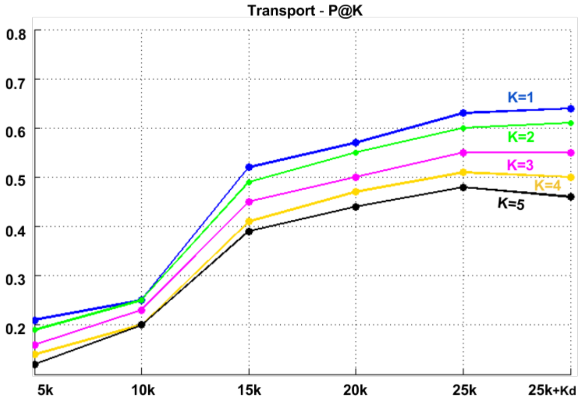


Figure 7.1. P@k scores for the `transport` domain.

False positives. We complement this experiment with a manual evaluation of *theoretical* false positives. Our intuition is that due to the nature of the task, some domains may be more flexible in allowing two terms to encode an *is-a* relation, while others may be more restrictive. We asked human judges to manually validate a sample of 200 *wrong pairs* from our best run in each domain, and estimated precision over them. As expected, *hard science* domains like `physics` obtain very low results (about 1% precision). In contrast, other domains like `education` (12% precision), or `transport` (16% precision), probably due to their multidisciplinary nature, allow more valid hypernyms for a given term than what is currently encoded in Wikidata.

7.3.5 Manual evaluation: Extra-coverage

In this experiment we evaluate the performance of TaxoEmbed on instances not included in Wikidata. We describe the experimental setting in Section 7.3.5 and present the results in Section 7.3.5.

Experimental setting

For this experiment we use two configurations of TaxoEmbed: the first one includes 25k domain-wise expanded training pairs (TaxE$_{25k}$), whereas the second one adds 1k pairs from KB-U (TaxE$_{25k+K^d}$). We randomly extract 200 test BabelNet synsets (20 per domain) whose hypernyms are missing in Wikidata. We compare against a number of taxonomy learning and Information Extraction systems, namely Yago (Suchanek et al., 2007a), WiBi (Flati et al., 2014) and DefIE (Delli Bovi et al., 2015b). Yago and WiBi are used as *upper bounds* due to the nature of their hypernymic relations. They include a great number of manually-encoded taxonomies (e.g. exploiting WordNet and Wikipedia categories). Yago derives its taxonomic relations from an automatic mapping between WordNet and Wikipedia categories. WiBi, on the other hand, exploits, among a number of different Wikipedia-specific heuristics, categories and the syntactic structure of the introductory sentence of Wikipedia pages. Finally, DefIE is an automaic OIE system relying on the syntactic structure of pre-disambiguated definitions (see Section 6.3.2 for more details).[29] Three annotators manually evaluated the validity of the hypernyms extracted by each system (one per test instance).

Results and discussion

Table 7.6 shows the results of TaxoEmbed and all comparison systems. As expected, Yago and WiBi achieve the best overall results. However, TaxoEmbed, based solely on distributional information, performed competitively in detecting new hypernyms when compared to DefIE, improving its recall in most domains, and even surpassing Yago in technical areas like `biology` or `health`. However, our model does not perform particularly well on `media` and `physics`. In most domains our model is able to discover novel hypernym relations that are not captured by any other system (e.g. *therapy* for *radiation treatment planning* in the `health` domain or *decoration* for *molding* in the `art` domain)[30].

In fact, the overlap between our approach and the remaining systems is actually quite small (on average less than 25% with all of them on the Extra-Coverage experiment). This is mainly due to the fact that TaxoEmbed only exploits distributional information and does not make use of predefined syntactic heuristics, suggesting that the information it provides and the rule-based comparison systems may be

[29] For this experiment, we included DefIE's *is-a* relations only.

[30] For simplicity, we use the word surface form to refer to BabelNet synsets.

	`art`			`biology`			`education`			`geography`			`health`		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TaxE$_{25k}$	0.45	0.45	0.45	**0.40**	**0.40**	**0.40**	**0.60**	**0.60**	**0.60**	0.35	0.35	0.35	**0.45**	**0.45**	**0.45**
TaxE$_{25k+K^d}$	0.50	**0.50**	**0.50**	**0.40**	**0.40**	**0.40**	0.55	0.55	0.55	0.35	0.35	0.35	**0.45**	**0.45**	**0.45**
DefIE	**0.63**	0.35	0.45	0.36	0.20	0.25	0.57	0.20	0.29	**0.66**	**0.40**	**0.50**	0.25	0.15	0.18
Yago	0.88	0.75	0.81	0.62	0.25	0.36	0.94	0.80	0.86	0.79	0.75	0.77	0.28	0.10	0.15
Wibi	0.70	0.70	0.70	0.58	0.50	0.54	0.94	0.80	0.86	0.75	0.75	0.75	0.66	0.50	0.57
	`media`			`music`			`physics`			`transport`			`warfare`		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TaxE$_{25k}$	0.10	0.10	0.10	0.45	0.45	0.45	0.15	**0.15**	0.15	0.35	**0.35**	0.35	0.25	0.25	0.25
TaxE$_{25k+K^d}$	0.10	0.10	0.10	0.40	0.40	0.40	0.15	**0.15**	0.15	0.25	0.25	0.25	0.45	**0.45**	**0.45**
DefIE	**0.81**	**0.45**	**0.58**	**0.71**	**0.50**	**0.58**	**0.42**	**0.15**	**0.22**	**0.54**	0.30	**0.38**	**0.60**	0.30	0.40
Yago	0.76	0.65	0.70	0.84	0.55	0.67	0.80	0.40	0.53	0.93	0.70	0.80	0.81	0.65	0.72
Wibi	0.90	0.90	0.90	0.89	0.85	0.87	0.68	0.55	0.61	0.87	0.70	0.77	0.66	0.50	0.57

Table 7.6. Precision, recall and F-Measure between TaxoEmbed, two taxonomy learning systems (Yago and WiBi), and a pattern-based approach that performs hypernym extraction (DefIE).

complementary. We foresee a potential avenue focused on combining a supervised distributional approach such as TaxoEmbed with syntactically-motivated systems such as Wibi or Yago. This combination of a distributional system and manual patterns was already introduced by Shwartz et al. (2016) on the hypernym detection task with highly encouraging results.

7.3.6 Conclusion

We have presented TaxoEmbed, a supervised taxonomy learning framework exploiting the property that was observed in Fu et al. (2014), namely that there exists, for a given domain-specific terminology, a shared linear projection among term-hypernym pairs. We showed how this can be used to learn a hypernym transformation matrix for discovering novel *is-a* relations, which are the backbone of lexical taxonomies. First, we allocate almost 2M BabelNet synsets into a predefined domain of knowledge. Then, we collect training data both from a manually constructed knowledge base (Wikidata), and from OIE systems. We substantially expand our initial training set by expanding both terms and hypernyms to all their available senses, and in a last step, to their corresponding disambiguated vector representations.

Evaluation shows that the general trend is that our hypernym matrix improves as we increase training data. Our best domain-wise configuration combines 25k training pairs from Wikidata and additional pairs from an OIE-derived KB, achieving promising results. The domains in which the addition of the OIE-based information contributed the most are `education`, `transport` and `media`. For instance, in the case of `education`, this may be due to the over representation of the North American educational system in IE systems like Nell. We accompany this quantitative evaluation with manual assessment of precision of false positives, and an analysis of the potential coverage comparing it with knowledge taxonomies like Yago or WiBi, and with DefIE, a *quasi*-OIE system.

For future work we are planning to apply this strategy to learn large-scale semantic relations beyond hypernymy. This may constitute a first step towards a global and fully automatic ontology learning system. In the context of semantic web, we would like to include semantic parsers and distant supervision to our algorithm in order to capture n-ary relations between pairs of concepts to further create and improve existing KBs. As mentioned in Section 7.3.5, we are also planning to combine our distributional approach with rule-based heuristics, following the line of work introduced by Shwartz et al. (2016). Finally, we see potential in the domain clustering approach for improving graph-based taxonomy learning systems, as it can serve as a weighting measure as to how pertinent a given set of concepts in a taxonomy are for a specific domain.

7.4 Collocation Discovery

The embedding of cues about how we perceive concepts and how these concepts relate and generalize across different domains gives knowledge resources the capacity of generalization, which lies at the core of human cognition (Yu et al., 2015) and is also central to many Natural Language Processing (NLP) applications (Jurgens and Pilehvar, 2015). It is general practice to identify and formalize conceptual relations using a reference knowledge repository. As such a repository, WordNet stands out as the *de facto* relational lexical database, containing over 200k English senses with 155k word forms (see Section 2.1). While the value of WordNet for NLP is indisputable, it is generally recognized that enriching it with additional information makes it an even more valuable resource. Thus, there is a line of research aimed at extending it with novel terminology (Jurgens and Pilehvar, 2016), cross-predicate relations (Lopez de la Calle et al., 2016), and so forth. Nonetheless, one type of information has been largely neglected so far: collocations, i.e., idiosyncratic binary lexical co-occurrences. As a standalone research topic, however, collocations have been the focus of a substantial amount of work, e.g. for automatically retrieving them from corpora (Choueka, 1988; Church and Hanks, 1989; Smadja, 1993; Kilgariff, 2006; Evert, 2007; Pecina, 2008; Bouma, 2010; Gao, 2013), and for their semantic classification according to different typologies (Wanner et al., 2006; Gelbukh and Kolesnikova., 2012; Moreno et al., 2013; Wanner et al., 2016). However, to the best of our knowledge, no previous work attempted the automatic enrichment of WordNet with collocational information. The only related attempt consisted in designing a schema for the manual inclusion of lexical functions from Explanatory Combinatorial Lexicology (Mel'čuk, 1996, ECL) into the Spanish EuroWordNet (Wanner et al., 2004).

Given the importance of collocations for a series of NLP applications (e.g. machine translation, text generation or paraphrasing), we propose to fill this gap by putting forward a new methodology which exploits intrinsic properties of state-of-the-art semantic vector space models and leverages the transformation matrix introduced by Mikolov et al. (2013b) in a word-level machine translation task. As a result, we release an extension of WordNet with detailed collocational information, named ColWordNet (CWN). This extension is carried out by means of the inclusion of *novel edges*, where each edge encodes a *collocates-with* relation, as well as the semantics of the collocation itself. For example, given the pair of synsets `desire.n.01` and `ardent.a.01`, a novel relation $\xrightarrow[x]{col:intense}$ is introduced, where 'intense' is the *semantic category* denoting *intensification*, and x is the confidence score assigned by our algorithm.

The remainder of this section is organized as follows: In Section 7.4.1, we provide some general background on collocations. Section 7.4.2 describes the methodology followed to construct CWN. Then, we evaluate CWN both intrinsically and extrinsically. Our intrinsic evaluation consists of a manual scoring of the correctness of the newly introduced relations (Section 7.4.3) while the extrinsic evaluation assesses the quality of CWN as an input resource for introducing collocational information into a word embeddings model (Section 7.4.4). Finally, Section 7.4.5 summarizes the main contributions of our work on collocation discovery and outlines potential avenues for future work.

7.4.1 Background

In what follows, we first present relevant background on the semantic categories of collocations used in our work. Collocations are restricted lexical co-occurrences of two syntactically related lexical items, the base and the collocate. In a collocation, the base is freely chosen by the speaker, while the choice of the collocate depends on the base; see, e.g., (Cowie, 1994; Mel'čuk, 1996; Kilgariff, 2006) for a theoretical discussion. For instance, in the collocations *take* [*a*] *step*, *solve* [*a*] *problem*, *pay attention*, *deep sorrow*, and *strong tea*, *step*, *problem*, *attention*, *sorrow* and *tea* are the bases and *take*, *solve*, *pay*, *deep* and *strong* their respective collocates.

Besides a syntactic dependency, between the base and the collocate a semantic relation holds. Some of these semantic relations, such as 'intense', 'weak', 'perform', 'cause', etc. can be found across a large number of collocations. For instance, an 'intense' *applause* is a *thundering applause*, an 'intense' *emotion* is *deep*, 'intense' *rain* is *heavy*, and so on. In our experiments, we focused on the subset of the most prominent eight semantic collocation relations (or categories), which are listed in the first column of Table 7.7. These semantic categories are a generalization of the *lexical functions* (LFs) from ECL already used in Wanner et al. (2004). We have decided to use somewhat more generic categories instead of LFs because, on the one hand, some of the LFs differ only in terms of their syntactic structure (i.e. they capture the same semantic relation), and, on the other hand, LFs pose a great challenge for annotation due to their syntactic granularity.

7.4.2 Methodology

In this section, we provide a detailed description of the algorithm behind the construction of CWN. The system takes as input the WordNet lexical database and a set of collocation lists pertaining to predefined semantic categories, and outputs CWN. First, we collect training data and perform automatic disambiguation. Then, we use this disambiguated data for training a linear *transformation matrix* from the base vector space to the collocate vector space. Finally, we exploit the WordNet taxonomy to select input base collocates to which we apply the transformation matrix in order to obtain a sorted list of candidate collocates.

Sem. Category	Example	# instances
‘intense’	*absolute certainty*	586
‘weak’	*remote chance*	70
‘perform’	*give chase*	393
‘begin to perform’	*take up a chase*	79
‘increase’	*improve concentration*	73
‘decrease’	*limit [a] choice*	73
‘create’, ‘cause’	*pose [a] challenge*	195
‘put an end’	*break [the] calm*	79

Table 7.7. Semantic categories and size of training set

Collecting and Disambiguating Training Data

As is common in previous work on semantic collocation classification (Moreno et al., 2013; Wanner et al., 2016), our training set consists of a list of manually annotated collocations. For this purpose, we randomly selected nouns from the Macmillan Dictionary and manually classified their corresponding collocates with respect to their semantic categories.[31] Note that there may be more than one collocate for each base. Since collocations with different collocate meanings are not evenly distributed in language (e.g., we may tend to use more often collocations conveying the idea of ‘intense’ and ‘perform’ than ‘begin to perform’), the number of instances per category in our training data also varies significantly (see Table 7.7).

Our training dataset consists at this stage of pairs of plain words, with the inherent ambiguity this gives raise to. We surmount this challenge by applying a disambiguation strategy based on the notion that, from all the available senses for a collocation’s base and collocate, their correct senses are those which are most similar. This is a strategy that has been proved effective in previous concept-level disambiguation tasks (Delli Bovi et al., 2015a). In this work we used SensEmbed (Iacobacci et al., 2015) as the base vector space (see Section 7.3.2 for a more detailed description of SensEmbed) and followed Delli Bovi et al. (2015a) to automatically disambiguate our training data using SensEmbed.

Formally, let us denote the SensEmbed vector space as $\mathcal{S}$, and our original text-based training data as $\mathbf{T}$. For each training collocation $\langle b, c\rangle \in \mathbf{T}$ we consider all the available lexicalizations (i.e., senses) for both the base b and the collocate c in $\mathcal{S}$, namely $L_b = \{l_b^1 ... l_b^n\}$, and $L_c = \{l_c^1 ... l_c^m\}$, and their corresponding set of sense embeddings $\mathbf{V}_b = \{\vec{v}_b^1, ..., \vec{v}_b^n\}$ and $\mathbf{V}_c = \{\vec{v}_c^1, ..., \vec{v}_c^m\}$. Our aim is to select, among all possible pairs of senses, the pair $\langle l'_b, l'_c\rangle$ that maximizes the cosine similarity between the corresponding embeddings v'_b and v'_c, which is computed as follows:

$$\langle \vec{v}'_b, \vec{v}'_c\rangle = \text{argmax}_{\vec{v}_b \in \mathbf{V}_b,\, \vec{v}_c \in \mathbf{V}_c} \frac{\vec{v}_b \cdot \vec{v}_c}{\|\vec{v}_b\| \, \|\vec{v}_c\|} \tag{7.4}$$

Our disambiguation strategy yields a set of disambiguated pairs $\mathbf{D}$. This is the input for the next module of the pipeline, the learning of a *transformation matrix* aimed at retrieving WordNet synset collocates for any given WordNet synset base.

[31] We do not consider phrasal verb collocates, e.g. *stand up*, *give up* or *calm down.*

Training a Sense-Level Transformation Matrix for each Semantic Category

Among the many properties of word embeddings that have been explored so far in the literature (e.g., modeling analogies or projecting similar words nearby in the vector space (Mikolov et al., 2013a,d)), the most pertinent to this work is the linear relation that holds between semantically similar words in two analogous spaces (Mikolov et al., 2013b). Mikolov et al.'s original work learned a linear projection between two monolingual embeddings models to train a word-level machine translation system between English and Spanish. Other examples include the exploitation of this property for language normalization, i.e. finding *regular English* counterparts of Twitter language (Tan et al., 2015), or hypernym discovery (see Section 7.3).

In our specific case, we learn a linear transformation from $\vec{v}'_b$ to $\vec{v}'_c$, aiming at reflecting an inherent condition of collocations. Since collocations are a linguistic phenomenon that is more frequent in the narrative discourse than in formal essays, they are less likely to appear in an encyclopedic corpus (recall that SensEmbed vectors are trained on a dump of the English Wikipedia). This motivates the use of $\mathcal{S}$ as our *base space*, and the SW2V word and synset vector space model (see Chapter 5) trained on the UMBC corpus (Han et al., 2013) as the *collocate model.* UMBC, which is a a corpus from paragraphs extracted from the web, contains a more diverse language, including for example blog posts and news items. Furthermore, we exploit the fact that SW2V represents both words and synsets in the same vector space for increasing the training data. Let us denote $\mathcal{X}$ the SW2V collocate vector space model.

Then, we construct our linear transformation model as follows: For each disambiguated collocation $\langle l'_b, l'_c \rangle \in \mathbf{D}$, we first retrieve the corresponding base vectors $\vec{v}'_b$. Next, we exploit the fact that $\mathcal{X}$ contains both BabelNet synsets and words, and derive for each l'_c two items, namely the vectors associated to its lexicalization (word-based) and its BabelNet synset. For example, for the training pair $\langle \texttt{ardent\_bn:00097467a}, \texttt{desire\_bn:00026551n} \rangle \in \mathbf{D}$, we learn two linear mappings, namely `ardent_bn:00097467a` $\mapsto$ `desire` and `ardent_bn:00097467a` $\mapsto$ `bn:00026551n`. We opt for this strategy, which doubles the size of the training data in most lexical functions (depending on coverage), due to the lack of resources of manually-encoded classification of collocations. By following this strategy we obtain an extended training set $\mathbf{D}* = \{\vec{b}_i, \vec{c}_i\}_{i=1}^n$ ($\vec{b}_i \in \mathcal{X}$, $\vec{c}_i \in \mathcal{S}, n \geq |\mathbf{D}|$). Then, we construct a *base matrix* $\mathbf{B} = \left[\vec{b}_1 \ldots \vec{b}_n\right]$ and a collocate matrix $\mathbf{C} = [\vec{c}_1 \ldots \vec{c}_n]$ with the resulting set of training vector pairs. We use these matrices to learn a linear transformation matrix $\Psi \in \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{X}}$, where $d_\mathcal{S}$ and and $d_\mathcal{X}$ are, respectively, the number of dimensions of the base vector space (i.e., SensEmbed and the collocate vector space SW2V.[32] Following the notation in Tan et al. (2015), this transformation can be depicted as:

$$\mathbf{B}\Psi \approx \mathbf{C}$$

As in Mikolov et al.'s original approach, the training matrix is learned by solving the following optimization problem:

[32] In our setting the numbers of dimensions are $d_\mathcal{S} = 400$ and $d_\mathcal{X} = 300$.

$$\min_{\Psi} \sum_{i=1}^{n} \| \Psi \vec{b}_i - \vec{c}_i \|^2$$

Having trained Ψ, the next step of the pipeline is to apply it over a subset of WordNet's base concepts and their hyponyms. For each synset in this branch, we apply a scoring and ranking procedure which assigns a *collocates-with* score. If such score is higher than a predefined threshold, tuned over a development set, this relation is included in CWN.

Retrieving and Sorting WordNet Collocate Synsets

During the task of enriching WordNet with collocational information, we first gather a set of base WordNet synsets by traversing WordNet hypernym hierarchy starting from those base concepts that are most fit for the input semantic category[33]. Then, the *transformation matrix* learned in Section 7.4.2 is used to find candidate WordNet synset collocates (mostly verbs or adjectives) for each base WordNet synset.

As explained in Section 7.4.2, WordNet synsets are mapped to BabelNet synsets, which in turn map to as many vectors in SensEmbed as their associated lexicalizations. Formally, given a base synset b, we apply the transformation matrix to all the SensEmbed vectors $\mathbf{V}_b = \{\vec{v}_b^1, ..., \vec{v}_b^n\}$ associated with its lexicalizations. For each $\vec{v}_b^i \in \mathbf{V}_b$, we first get the vector $\vec{\psi}_b^i = \vec{v}_b^i \Psi$ obtained as a result of applying the transformation matrix and then we gather the subset $W_b^i = \{\vec{w}_b^{i,1} \dots \vec{w}_b^{i,10}\}$ ($\vec{w}_b^{i,j} \in \mathcal{X}$) of the top ten closest vectors by cosine similarity to $\vec{\psi}_b^i$ in the SW2V vector space $\mathcal{X}$. Each $\vec{w}_b^{i,j}$ is ranked according to a scoring function $\lambda(\cdot)$, which is computed as follows[34]: $\lambda(\vec{w}_b^{i,j}) = \frac{cos(\vec{\psi}_b^i, \vec{w}_b^{i,j})}{j}$. This scoring function takes into account both the cosine similarity as well as the relative position[35] of the candidate collocate with respect to other neighbors in the vector space. Apart from sorting the list of candidate collocates, this scoring function is also used to measure the confidence of the retrieved collocate synsets in CWN.

7.4.3 Intrinsic evaluation: Precision of collocate relations

Sampling and evaluation are carried out as follows. First, for each semantic category, we retrieve 50 random bases included in the aforementioned base concepts (see Section 7.4.2) and all their hyponym branch. This results in an evaluation set $Test$ of 800 collocations, as for each base we retrieve the 5 highest scoring candiates. These collocations are evaluated in terms of correctness, i.e., if the associated synset is an appropriate collocate for the input base. Note that not all bases in the test set may be suitable for the given semantic category, and that is why we also perform an evaluation on the test data restricted to only those bases manually selected for being suitable for having at least one collocate. We denote the restricted test data

[33] These are: For 'intense' and 'weak', `attitude.n.01`, `feeling.n.01` and `ability.n.02`. For the rest of them, we select `cognition.n.01`, `act.n.02` and `action.n.01`.

[34] If $w_b^{i,j}$ appears in a different W_b^j set ($j \neq i$), its scores are averaged.

[35] Position is arguably an important factor as there may be dense areas where cosine similarity alone may not reflect entirely the fitness of a candidate.

	'intense'				**'perform'**				**'put an end'**				**'increase'**			
	Baseline		CWN		Baseline		CWN		Baseline		CWN		Baseline		CWN	
	Test	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**
P@1	0.00	0.00	0.35	**0.46**	0.15	0.16	0.20	**0.36**	0.05	0.08	0.15	**0.50**	0.05	0.14	0.15	**0.42**
P@5	0.03	0.30	0.43	**0.57**	0.06	0.06	0.13	**0.23**	0.02	0.03	0.12	**0.40**	0.04	0.11	0.18	**0.51**
MRR	0.05	0.41	0.48	**0.65**	0.18	0.19	0.32	**0.59**	0.07	0.12	0.20	**0.68**	0.07	0.21	0.22	**0.65**
MAP	0.05	0.45	0.48	**0.64**	0.15	0.18	0.32	**0.59**	0.07	0.12	0.19	**0.64**	0.07	0.20	0.22	**0.64**
	'decrease'				**'create/cause'**				**'weak'**				**'begin to perform'**			
	Baseline		CWN		Baseline		CWN		Baseline		CWN		Baseline		CWN	
	Test	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**
P@1	0.00	0.00	0.30	**0.46**	0.05	0.16	0.10	**0.50**	0.00	0.00	0.10	**0.22**	0.00	0.00	0.00	0.00
P@5	0.02	0.03	0.19	**0.29**	0.04	0.13	0.04	**0.20**	0.02	0.03	0.04	**0.08**	0.03	0.07	0.02	**0.20**
MRR	0.02	0.04	0.39	**0.61**	0.07	0.25	0.10	**0.50**	0.03	0.04	0.01	**0.22**	0.05	0.12	0.04	**0.41**
MAP	0.02	0.03	0.38	**0.58**	0.06	0.20	0.10	**0.50**	0.03	0.04	0.01	**0.22**	0.05	0.12	0.04	**0.41**

Table 7.8. Manual evaluation of the performance of ColWordNet.

as $Test*$. For example, the base synset `putt.n.01` defined as *hitting a golf ball that is on the green using a putter* does not admit any 'decrease' collocate, and therefore its collocations are not considered in $Test*$.

Since our algorithm returns a list of candidate collocate synsets for an input base synset, the task naturally becomes that of a ranking problem, and therefore ranking metrics such as Precision@K (P@K), Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) are appropriate for evaluating this experiment. These measures provide insights on different aspects of the outcome of the task, e.g. how often valid collocates were retrieved in the first positions of the rank (MRR), and if there were more than one valid collocate, whether this set was correctly retrieved, (MAP and R-P)[36]. In Table 7.8 we provide a detailed summary of the performance of our system (CWN), as compared with a competitor unsupervised baseline which exploits word analogies (as in $\vec{man} - \vec{king} + \vec{woman} = \vec{queen}$). This baseline, which we deploy on the SW2V space, takes as input a prototypical collocation of a given semantic category (e.g. *thunderous applause* for 'intense') and an input base, and collects the top 10 Nearest Neighbours (NNs) to the vector resulting of the aforementioned analogy operation. This approach was recently used in a similar setting (Rodríguez-Fernández et al., 2016). Due to the difficulty of the task, and the restriction it imposes for collocates to be disambiguated synsets rather than any text-based word, the unsupervised approach fails short when compared to our supervised method, which is capable to find more and better disambiguated collocates.

Note that for half of the semantic categories under evaluation, our approach correlated well with human judgement, with the highest ranking candidates being more often correct than those ranked lower. This is the case of 'put an end', 'decrease', 'create/cause' and 'weak'. In fact, it is in 'put an end', where our system achieves the

[36] See Bian et al. (2008) for an in-depth analysis of these metrics.

highest MRR score, which we claim to be the most relevant measure, as it rewards cases where the first ranked returned collocation is correct without measuring in the retrieved collocates at other positions. Moreover, let us highlight the importance of two main factors. First, the need for a well-defined semantic relation between bases and collocates. It has been shown in other tasks that exploit linear transformations between embeddings models that even for one single relation there may be clusters that require certain specificity in the *domain* or *semantic* of the data (see Fu et al. (2014) for a discussion of this phenomenon in the task of taxonomy learning). Second, the importance of having a reasonable amount of training pairs so that the model can learn the idiosyncrasies of the semantic relation that is being encoded (e.g., Mikolov et al. (2013b) report a major increase in performance as training data increases in several orders of magnitude). This is reinforced in our experiments, where we obtain the highest MAP score for 'intense', the semantic category for which we have the largest training data available.

7.4.4 Extrinsic evaluation: Retrofitting vector space models to ColWordNet

We complement our manual evaluation with an extrinsic experiment, where we assess the extent to which our newly generated lexical resource can be used to *introduce collocational sensitivity* to a generic word embeddings model[37]. To this end, we extract collocation clusters by extracting all the synsets associated lemmas (e.g. for *heavy.a.01 rain.n.01*, we would extract the cluster [*heavy*, *rain*, *rainfall*]). These are used as input for the Retrofitting word vectors algorithm (Faruqui et al., 2015)[38]. This algorithm takes as input a vector space and a semantic lexicon which may encode any semantic relation, and puts closer in the vector space words that are related in the lexicon.

Previous approaches have encoded semantic relations by introducing some kind of bias into a vector space model (Yu et al., 2015; Pham et al., 2015; Nguyen et al., 2016; Mrkšic et al., 2017). For instance, Yu et al. (2015) encode (term, hypernym) relations by grouping together terms and their hypernyms, rather than semantically related items. In this way, their *biased* model puts closer to *jaguar* terms like *animal* or *vehicle*, while an unbiased model would put nearby terms such as *lion*, *bmw* or *jungle*. We aim at introducing a similar bias, but in terms of collocational information. This is achieved, for each lexical function and each synset in CWN-*st*, by obtaining its top 3 collocate candidates and incorporate information on their *collocationality* into the model.

Collocational sensitivity

In this experiment, we assess the extent to which a retrofitted model with collocational bias is able to discriminate between a correct collocation and a random combination of the same base with an unrelated collocate. To this end, we manually constructed two datasets, one for *noun+adjective* ('intense' and 'weak' semantic categories) and one

[37] We use the Google News pre-trained Word2Vec vectors, available at `code.google.com/archive/p/Word2Vec/`, as input for retrofitting.

[38] We used the code available at `https://github.com/mfaruqui/retrofitting`

	‘intense’			**‘weak’**			**‘perform’**			**‘create/cause’**		
	correct	*dist.*	*diff.*	*correct*	dist.	*diff.*	*correct*	*dist.*	*diff.*	*correct*	*dist.*	*diff.*
original	0.22	0.04	+0.18	0.17	0.05	+0.12	0.15	0.05	+0.10	0.17	0.06	+0.11
retrofitted	0.27	0.06	**+0.21**	0.19	0.06	**+0.13**	0.25	0.11	**+0.14**	0.28	0.12	**+0.16**

Table 7.9. Comparison of collocational sensitivity between original and retrofitted embeddings models over four semantic categories.

for *noun+verb* combinations, which we evaluate on the two most productive semantic categories, namely ‘perform’ and ‘create/cause’. The datasets consist of 50 bases and one of their correct collocates according to the Macmillan Collocations Dictionary, accompanied by four *distractor* (*dist.* in Table 7.9) collocates. For instance, given the correct ‘perform’ collocation *make a pledge*, we expect our ‘perform’-wise retrofitted model to increase the score in $\vec{make} + \vec{pledge}$ substantially more than a combination $\vec{pledge} + \vec{distractor}$. For each evaluated semantic category, we computed the average increase of the cosine similarity between all correct collocations and all distractors (*diff.* in Table 7.9). As shown in Table 7.9, there is a consistent increase over the four evaluated semantic categories, namely ‘intense’, ‘weak’, ‘perform’ and ‘create/cause’. This proves the potential of our retrofitted model to discern between correct and wrong collocates. In the following section, we explore the possibility to use this vector space for finding collocates giving a base as input.

Exploring Nearest Neighbours for collocate discovery

Inspired by Yu et al. (2015) work on introducing hypernymic bias into a word embeddings model, we explore the extent to which our retrofitted models can be used to discover *alternative collocates* given the composition of the words involved in a collocation as input. In order to discover these collocates, we compose the base and the collocate by averaging their respective word embeddings and retrieve its closest words in the vector space according to cosine similarity. In Table 7.10 we show a sample of five NNs for several input adjective+noun collocations for the ‘intense’ semantic category. These examples reveal how the vector space model retroffited using our collocations tends to bring closer in the space modifiers (i.e., collocates), providing an interesting method for automatic collocation discovery. Despite its simplicity, this collocational discovery approach extracts a considerable amount of suitable fine-grained collocates for a given base. For example, given the collocation *intense sympathy*, the retrofitted space extracts *considerable*, *tremendous*, *enormous* and *immense* as candidate collocates of intensity among the five nearest neighbours. As future work we plan to further exploit and evaluate the impact of this property.

7.4.5 Conclusion

We have described a system for an automatic enrichment of the WordNet lexical database with fine-grained collocational information, yielding a resource called ColWordNet (CWN). Our approach is based on the intuition that there is a linear transformation in vector spaces between bases and collocates of the same *semantic*

	‘**intense**’	
	`original`	`retrofitted`
ferocious + hatred	**vicious**	fierce
	fury	**fearsome**
	ferocity	fury
	savage	hate
	hostility	**savage**
intense + sympathy	fierce	**considerable**
	empathy	**tremendous**
	admiration	**enormous**
	anger	encouragement
	grudging respect	**immense**
sheer + delight	amazement	**immense**
	sheer unadulterated	**colossal**
	sheer joy	delectation
	joy	disgust
	astonishment	**stupendous**

Table 7.10. Comparison of the five NNs of six sample adj+noun collocations between a generic word embeddings model and a *retrofitted* version with semantic collocation information (‘intense’). Note the increase in plausible collocates in retrofitted models (in bold).

category, e.g. between *heavy* and *rain*, or between *ardent* and *desire*. We have exploited sense-based embedding models to train an algorithm designed to retrieve valid collocates for a given input base. This pipeline is carried out at the *sense* level (rather than the word level), by leveraging sense embedding models which use BabelNet as a reference sense inventory (Iacobacci et al., 2015; Mancini et al., 2017). In particular, the flexibility of SW2V, trained on a corpus extracted from the web, proved crucial on the final result. We evaluated CWN both intrinsically and extrinsically, and verified that our algorithm is able to accurately encode fine-grained *collocates-with* relations at the synset level.

In the future, we plan to design a method to retrieve the best *bases* for a given semantic category, which would allow us not to rely on predefined manually built base concepts. Finally, we are currently investigating the potential of applying neural approaches recasting the task as a sequence classification problem for including collocational information in WordNet clusters.

Chapter 8

Downstream NLP Applications: Text Categorization and Sentiment Analysis

As a general trend, most current Natural Language Processing (NLP) systems function at the word level, i.e. individual words constitute the most fine-grained meaning-bearing elements of their input. The word level functionality can affect the performance of these systems in two ways: (1) it can hamper their efficiency in handling words that are not encountered frequently during training, such as multiwords, inflections and derivations, and (2) it can restrict their semantic understanding to the level of words, with all their ambiguities, and thereby prevent accurate capture of the intended meanings. The first issue has recently been alleviated by techniques that aim to boost the generalisation power of NLP systems by resorting to sub-word or character-level information (Ballesteros et al., 2015; Kim et al., 2016). The second limitation, however, has not yet been studied sufficiently. In in order to deal with these issues simultaneously we propose a sense-based pipeline which can be seamlessly integrated into any neural architecture.

We evaluate the pipeline in two downstream NLP applications: polarity detection and topic categorization. Specifically, we use a classification model based on Convolutional Neural Networks which has been shown to be very effective in various text classification tasks (Kalchbrenner et al., 2014; Kim, 2014; Johnson and Zhang, 2015; Tang et al., 2015; Xiao and Cho, 2016). We show that a simple disambiguation of input can lead to performance improvement of a state-of-the-art text classification system on multiple datasets, particularly for long inputs and when the granularity of the sense inventory is reduced. Our pipeline is quite flexible and modular, as it permits the integration of different WSD and sense representation techniques.

8.1 Related Work

Despite the various studies on sense representation learning (see Section 3.2), the integration of sense representations into deep learning models has not been so straightforward, and research in this field has often opted for alternative evaluation benchmarks such as WSD, or artificial tasks, such as word similarity. Consequently, the problem of integrating sense representations into downstream NLP applications has remained understudied, despite the potential benefits it can have. Li and Jurafsky (2015) proposed a “multi-sense embedding" pipeline to check the benefit that can be gained by replacing word embeddings with sense embeddings in multiple tasks. With the help of two simple disambiguation algorithms, unsupervised sense embeddings were integrated into various downstream applications, with varying degrees of success. Given the interdependency of sense representation and disambiguation in this model, it is very difficult to introduce alternative algorithms into its pipeline, either to benefit from the state of the art, or to carry out an evaluation.

Instead, our pipeline provides the advantage of being modular: thanks to its use of disambiguation in the pre-processing stage and use of sense representations that are linked to external sense inventories, different WSD techniques and sense representations can be easily plugged in and checked. Along the same lines, Flekova and Gurevych (2016) proposed a technique for learning supersense representations, using automatically-annotated corpora. Coupled with a supersense tagger, the representations were fed into a neural network classifier as additional features to the word-based input. Through a set of experiments, Flekova and Gurevych (2016) showed that the supersense enrichment can be beneficial to a range of binary classification tasks. Our proposal is different in that it focuses directly on the benefits that can be gained by semantifying the input, i.e. reducing lexical ambiguity in the input text, rather than assisting the model with additional sources of knowledge. An input text is transformed from its surface-level semantics to the deeper level of word senses, i.e. their intended meanings. We take a step in this direction by designing a pipeline that enables seamless integration of word senses into downstream NLP applications, while benefiting from knowledge extracted from semantic networks. To this end, we propose a quick graph-based Word Sense Disambiguation (WSD) algorithm which allows high confidence disambiguation of words without much computation overload on the system.

8.2 Disambiguation Algorithm

Our proposal relies on a seamless integration of word senses in word-based systems. The goal is to semantify the text prior to its being fed into the system by transforming its individual units from word surface form to the deeper level of word senses. The semantification step is mainly tailored towards resolving ambiguities, but it brings about other advantages mentioned in the previous section. The aim is to provide the system with an input of reduced ambiguity which can facilitate its decision making.

To this end, we developed a simple graph-based joint disambiguation and entity linking algorithm which can take any arbitrary semantic network as input. The gist of our disambiguation technique lies in its speed and scalability. Conventional knowledge-based disambiguation systems (Hoffart et al., 2012; Agirre et al., 2014; Moro et al., 2014; Ling et al., 2015; Pilehvar and Navigli, 2014b) often rely on computationally expensive graph algorithms, which limits their application to on-the-fly processing of large number of text documents, as is the case in our experiments. Moreover, unlike supervised WSD and entity linking techniques (Zhong and Ng, 2010; Cheng and Roth, 2013; Melamud et al., 2016; Limsopatham and Collier, 2016), our

Algorithm 3 Disambiguation algorithm

```
Input: Input text $T$ and semantic network $N$
Output: Set of disambiguated senses $\hat{S}$
 1: Graph representation of $T$: $(S, E) \leftarrow \text{getGraph}(T, N)$
 2: $\hat{S} \leftarrow \emptyset$
 3: for each iteration $i \in \{1, ..., len(T)\}$
 4:     $\hat{s} = \text{argmax}_{s \in S} |\{(s, s') \in E : s' \in S\}|$
 5:     $\text{maxDeg} = |\{(\hat{s}, s') \in E : s' \in S\}|$
 6:     if $\text{maxDeg} < \theta|S| \,/\, 100$ then
 7:         break
 8:     else
 9:         $\hat{S} \leftarrow \hat{S} \cup \{\hat{s}\}$
10:         $E \leftarrow E \setminus \{(s, s') : s \vee s' \in \text{getLex}(\hat{s})\}$
11: return Disambiguation output $\hat{S}$
```

algorithm relies only on semantic networks and does not require any sense-annotated data, which is limited to English and almost non-existent for other languages.

Algorithm 3 shows our procedure for disambiguating an input document T. First, we retrieve from our semantic network the list of candidate senses[1] for each content word, as well as semantic relationships among them. As a result, we obtain a graph representation (S, E) of the input text, where S is the set of candidate senses and E is the set of edges among different senses in S. The graph is, in fact, a small sub-graph of the input semantic network, N. Our algorithm then selects the best candidates iteratively. In each iteration, the candidate sense that has the highest graph degree maxDeg is chosen as the winning sense:

$$\text{maxDeg} = \max_{s \in S} |\{(s, s') \in E : s' \in S\}| \tag{8.1}$$

After each iteration, when a candidate sense $\hat{s}$ is selected, all the possible candidate senses of the corresponding word (i.e. getLex($\hat{s}$)) are removed from E (line 10 in the algorithm). Note that this algorithm is in essence very similar to the one proposed in SW2V to connect word and senses in context (Section 5.1). The main difference lies on this last step of removing candidates after each iteration, which is targeted to reduce noise on the constructed graph.

Figure 8.1 shows a simplified version of the graph for a sample sentence. The algorithm would disambiguate the content words in this sentence as follows. It first associates *Oasis* with its *rock band* sense, since its corresponding node has the highest degree, i.e. 3. On the basis of this, the *desert* sense of *Oasis* and its link to the *stone* sense of *rock* are removed from the graph. In the second iteration, *rock band* is disambiguated as *music band* given that its degree is 2.[2] Finally, *Manchester* is associated with its *city* sense (with a degree of 1).

In order to enable disambiguating at different confidence levels, we introduce a threshold θ which determines the stopping criterion of the algorithm. Iteration continues until the following condition is fulfilled: $\text{maxDeg} < \theta|S| \,/\, 100$. This ensures

[1] As defined in the underlying sense inventory, up to trigrams. We used Stanford CoreNLP (Manning et al., 2014) for tokenization, Part-of-Speech (PoS) tagging and lemmatization.

[2] For bigrams and trigrams whose individual words might also be disambiguated (such as *rock* and *band* in *rock band*), the longest unit has the highest priority (i.e. *rock band*).

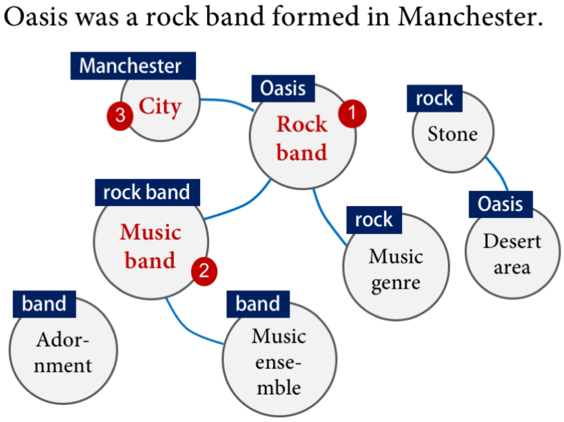


Figure 8.1. Simplified graph-based representation of a sample sentence.

that the system will only disambiguate those words for which it has a high confidence and backs off to the word form otherwise, avoiding the introduction of unwanted noise in the data for uncertain cases or for word senses that are not defined in the inventory.

8.3 Classification Model

In our experiments, we use a standard neural network based classification approach which is similar to the Convolution Neural Network classifier of Kim (2014) and the pioneering model of Collobert et al. (2011). Figure 8.2 depicts the architecture of the model. The network receives the concatenated vector representations of the input words, $\mathbf{v}_{1:n} = \mathbf{v}_1 \oplus \mathbf{v}_2 \oplus \cdots \oplus \mathbf{v}_n$, and applies (convolves) filters F on windows of h words, $m_i = f(\mathrm{F}.\mathbf{v}_{i:i+h-1} + b)$, where b is a bias term and $f()$ is a non-linear function, for which we use ReLU (Nair and Hinton, 2010). The convolution transforms the input text to a feature map $m = [m_1, m_2, \ldots, m_{n-h+1}]$. A max pooling operation then selects the most salient feature $\hat{m} = \max\{m\}$ for each filter.

In the network of Kim (2014), the pooled features are directly passed to a fully connected softmax layer whose outputs are class probabilities. However, we add a recurrent layer before softmax in order to enable better capturing of long-distance dependencies. It has been shown by Xiao and Cho (2016) that a recurrent layer can replace multiple layers of convolution and be beneficial, particularly when the length of input text grows. Specifically, we use a Long Short-Term Memory (Hochreiter and Schmidhuber, 1997, LSTM) as our recurrent layer which was originally proposed to avoid the vanishing gradient problem and has proven its abilities in capturing distant dependencies. The LSTM unit computes three gate vectors (forget, input, and output) as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \; g_t + \mathbf{U}_f \; h_{t-1} + \mathbf{b}_f), \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \; g_t + \mathbf{U}_i \; h_{t-1} + \mathbf{b}_i), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \; g_t + \mathbf{U}_o \; h_{t-1} + \mathbf{b}_o),
\end{aligned} \tag{8.2}$$

where $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{b}$ are model parameters and g and h are input and output sequences,

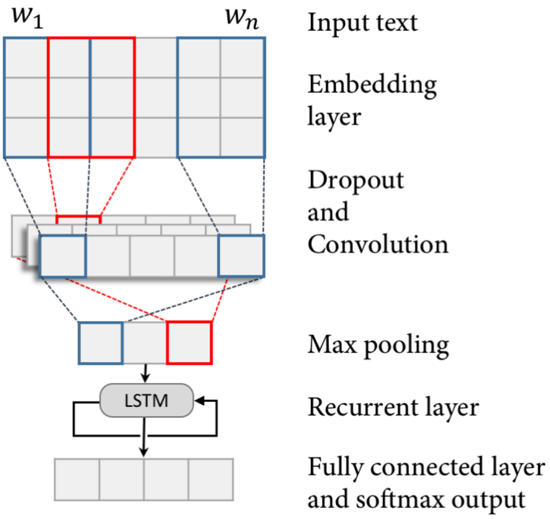


Figure 8.2. Text classification model architecture.

respectively. The cell state vector $\mathbf{c}_t$ is then computed as $\mathbf{c}_t = \mathbf{f}_t\ \mathbf{c}_{t-1} + \mathbf{i}_t\ \tanh(\tilde{\mathbf{c}}_t)$ where $\tilde{c}_t = \mathbf{W}_c\ g_t + \mathbf{U}_c\ h_{t-1}$. Finally, the output sequence is computed as $h_t = \mathbf{o}_t\ \tanh(\mathbf{c}_t)$. As for regularization, we used dropout (Hinton et al., 2012) after the embedding layer.

We perform experiments with two configurations of the embedding layer: (1) *Random*, initialized randomly and updated during training, and (2) *Pre-trained*, initialized by pre-trained representations and updated during training. In the following section we describe the pre-trained word and sense representation used for the initialization of the second configuration.

8.3.1 Pre-trained Word and Sense Embeddings

One of the main advantages of neural models is that they usually represent the input words as dense vectors. This can significantly boost a system's generalisation power and results in improved performance (Zou et al., 2013; Bordes et al., 2014; Kim, 2014; Weiss et al., 2015, *interalia*). This feature also enables us to directly plug in pre-trained sense representations and check them in a downstream application.

In our experiments we generate a set of sense embeddings by extending DeConf, a recent technique with state-of-the-art performance on multiple semantic similarity benchmarks (Pilehvar and Collier, 2016). We leave the evaluation of other representations to future work. DeConf gets a pre-trained set of word embeddings and computes sense embeddings in the same semantic space. To this end, the approach exploits the semantic network of WordNet (see Section 2.1), using the Personalized PageRank (Haveliwala, 2002) algorithm, and obtains a set of *sense biasing words* $\mathcal{B}_s$ for a word sense s. The sense representation of s is then obtained using the following formula:

$$\hat{\mathbf{v}}(s) = \frac{1}{|\mathcal{B}_s|} \sum_{i=1}^{|\mathcal{B}_s|} \mathrm{e}^{\frac{-i}{\delta}}\,\mathbf{v}(w_i), \tag{8.3}$$

where δ is a decay parameter and $\mathbf{v}(w_i)$ is the embedding of w_i, i.e. the i^{th} word in the sense biasing list of s, i.e. $\mathcal{B}_s$. We follow Pilehvar and Collier (2016) and set $\delta = 5$. Finally, the vector for sense s is calculated as the average of $\hat{\mathbf{v}}(s)$ and the embedding of its corresponding word.

Owing to its reliance on WordNet's semantic network, DeConf is limited to generating only those word senses that are covered by this lexical resource. We propose to use Wikipedia, and more concretely NASARI lexical vectors, in order to expand the vocabulary of the computed word senses. Wikipedia provides a high coverage of named entities and domain-specific terms in many languages, while at the same time also benefiting from a continuous update by collaborators. Moreover, it can easily be viewed as a sense inventory where individual articles are word senses arranged through hyperlinks and redirections. Recall from Chapter 4 that NASARI lexical vectors were composed of words with their respective weights. We view these lists as *biasing words* for individual Wikipedia pages, and then leverage the exponential decay function (Equation 8.3) to compute new sense embeddings in the same semantic space. In order to represent both WordNet and Wikipedia sense representations in the same space, we rely on the WordNet-Wikipedia mapping provided by BabelNet (see Section 2.3). For the WordNet synsets which are mapped to Wikipedia pages in BabelNet, we average the corresponding Wikipedia-based and WordNet-based sense embeddings.

8.3.2 Pre-trained Supersense Embeddings

It has been argued that WordNet sense distinctions are too fine-grained for many NLP applications (Hovy et al., 2013). The issue can be tackled by grouping together similar senses of the same word, either using automatic clustering techniques (Navigli, 2006; Agirre and Lopez, 2003; Snow et al., 2007) or with the help of WordNet's lexicographer files[3]. Various applications have been shown to improve upon moving from senses to supersenses (Rüd et al., 2011; Severyn et al., 2013; Flekova and Gurevych, 2016). In WordNet's lexicographer files there are a total of 44 sense clusters, referred to as supersenses, for categories such as *event*, *animal*, and *quantity*. In our experiments we use these supersenses in order to reduce granularity of our WordNet and Wikipedia senses. To generate supersense embeddings, we simply average the embeddings of senses in the corresponding cluster.

8.4 Evaluation

We evaluated our model on two classification tasks: topic categorization (Section 8.4.2) and polarity detection (Section 8.4.3). In the following section we present the common experimental setup.

8.4.1 Experimental setup

Classification model. Throughout all the experiments we used the classification model described in Section 8.3. The general architecture of the model was the

[3] https://wordnet.princeton.edu/man/lexnames.5WN.html

same for both tasks, with slight variations in hyperparameters given the different natures of the tasks, following the values suggested by Kim (2014) and Xiao and Cho (2016) for the two tasks. Hyperparameters were fixed across all configurations in the corresponding tasks. The embedding layer was fixed to 300 dimensions, irrespective of the configuration, i.e. Random and Pre-trained. For both tasks the evaluation was carried out by 10-fold cross-validation unless standard training-testing splits were available. The disambiguation threshold θ (cf. Section 8.2) was tuned on the training portion of the corresponding data, over seven values in [0,3] in steps of 0.5.[4] We used Keras (Chollet, 2015) and Theano (Team, 2016) for our model implementations.

Semantic network. The integration of senses was carried out as described in Section 8.2. For disambiguating with both WordNet and Wikipedia senses we relied on the joint semantic network of Wikipedia hyperlinks and WordNet via the mapping provided by BabelNet.[5]

Pre-trained word and sense embeddings. Throughout all the experiments we used Word2vec (Mikolov et al., 2013a) embeddings, trained on the Google News corpus.[6] We truncated this set to its 250K most frequent words. We also used WordNet 3.0 and the Wikipedia dump of November 2014 to compute the sense embeddings (see Section 8.3.1). As a result, we obtained a set of 757,262 sense embeddings in the same space as the pre-trained Word2vec word embeddings. We used DeConf (Pilehvar and Collier, 2016) as our pre-trained WordNet sense embeddings. All vectors had a fixed dimensionality of 300.

Supersenses. In addition to WordNet senses, we experimented with supersenses (see Section 8.3.2) to check how reducing granularity would affect system performance. For obtaining supersenses in a given text we relied on our disambiguation pipeline and simply clustered together senses belonging to the same WordNet supersense.

Evaluation measures. We report the results in terms of standard accuracy and F1 measures.[7]

8.4.2 Topic Categorization

The task of topic categorization consists of assigning a label (i.e. topic) to a given document from a pre-defined set of labels.

[4] We observed that values higher than 3 led to very few disambiguations. While the best results were generally achieved in the [1.5,2.5] range, performance differences across threshold values were not statistically significant in most cases.

[5] For simplicity we refer to this joint sense inventory as Wikipedia, but note that WordNet senses are also covered.

[6] `https://code.google.com/archive/p/Word2Vec/`

[7] Since all models in our experiments provide full coverage, accuracy and F1 denote micro- and macro-averaged F1, respectively Yang (1999).

Dataset	Domain	#classes	#docs	Avg doc size	Vocab size	Coverage	Evaluation
BBC	News	5	2,225	439.5	35,628	87.4%	10 cross valid.
Newsgroups	News	6	18,846	394.0	225,046	83.4%	Train-Test
Ohsumed	Medical	23	23,166	201.2	65,323	79.3%	Train-Test

Table 8.1. Statistics of the topic categorization datasets.

Datasets

For this task we used two newswire and one medical topic categorization datasets. Table 8.1 summarizes the statistics of each dataset.[8] The **BBC news** dataset[9] (Greene and Cunningham, 2006) comprises news articles taken from BBC, divided into five topics: business, entertainment, politics, sport and tech. **Newsgroups** (Lang, 1995) is a collection of 11,314 documents for training and 7532 for testing[10] divided into six topics: computing, sport and motor vehicles, science, politics, religion and sales.[11] Finally, **Ohsumed**[12] is a collection of medical abstracts from MEDLINE, an online medical information database, categorized according to 23 cardiovascular diseases. For our experiments we used the partition split of 10,433 documents for training and 12,733 for testing.[13]

Results

Table 8.2 shows the results of our classification model and its variants on the three datasets.[14] When the embedding layer is initialized randomly, the model integrated with word senses consistently improves over the word-based model, particularly when the fine-granularity of the underlying sense inventory is reduced using supersenses (with statistically significant gains on the three datasets). This highlights the fact that a simple disambiguation of the input can bring about performance gain for a state-of-the-art classification system. Also, the better performance of supersenses suggests that the sense distinctions of WordNet are too fine-grained for the topic categorization task. However, when pre-trained representations are used to initialize the embedding layer, no improvement is observed over the word-based model. This can be attributed to the quality of the representations, as the model utilizing them was unable to benefit from the advantage offered by sense distinctions. Our results suggest that research in sense representation should put special emphasis on real-world evaluations on benchmarks for downstream applications, rather than on artificial tasks such as word similarity. In fact, research has previously shown that

[8] The coverage of the datasets was computed using the 250K top words in the Google News Word2vec embeddings.

[9] `http://mlg.ucd.ie/datasets/bbc.html`

[10] We used the train-test partition available at `http://qwone.com/~jason/20Newsgroups/`

[11] The dataset has 20 fine-grained categories clustered into six general topics. We used the coarse-grained labels for their clearer distinction and consistency with BBC topics.

[12] `ftp://medir.ohsu.edu/pub/ohsumed`

[13] `http://disi.unitn.it/moschitti/corpora.htm`

[14] Symbols * and † indicate the sense-based model with the smallest margin to the word-based model whose accuracy is statistically significant at 0.95 confidence level according to unpaired t-test (* for positive and † for negative change).

Initialization	**Input type**		**BBC News**		**Newsgroups**		**Ohsumed**	
			Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
Random	**Word**		93.0	92.8	87.7	85.6	30.1	20.7
	Sense	**WordNet**	**93.5**	**93.3**	**88.1**	**86.9**	27.2†	18.3
		Wikipedia	92.7	92.5	86.7	84.9	29.7	**20.9**
	Supersense	**WordNet**	**93.6**	**93.4**	**90.1***	**89.0**	**31.8***	**22.0**
		Wikipedia	**94.6***	**94.4**	**88.5**	**85.8**	**31.1**	**21.3**
Pre-trained	**Word**		97.6	97.5	91.1	90.6	29.4	20.1
	Sense	**WordNet**	97.3	97.1	90.2	88.6	**30.2**	**20.4**
		Wikipedia	96.3	96.2	89.6†	88.9	**32.4**	**22.3**
	Supersense	**WordNet**	96.8	96.7	89.6	88.9	**29.5**	19.9
		Wikipedia	96.9	96.9	88.6	87.4	**30.6***	**20.3**

Table 8.2. Classification performance at the word, sense, and supersense levels with random and pre-trained embedding initialization. We show in bold those settings that improve the word-based model.

word similarity might not constitute a reliable proxy to measure the performance of word embeddings in downstream applications (Tsvetkov et al., 2015; Chiu et al., 2016).

Among the three datasets, Ohsumed proves to be the most challenging one, mainly for its larger number of classes (i.e. 23) and its domain-specific nature (i.e. medicine). Interestingly, unlike for the other two datasets, the introduction of pre-trained word embeddings to the system results in reduced performance on Ohsumed. This suggests that general domain embeddings might not be beneficial in specialized domains, which corroborates previous findings by Yadav et al. (2017) on a different task, i.e. entity extraction. This performance drop may also be due to diachronic issues (Ohsumed dates back to the 1980s) and low coverage: the pre-trained Word2vec embeddings cover 79.3% of the words in Ohsumed (see Table 8.1), in contrast to the higher coverage on the newswire datasets, i.e. Newsgroups (83.4%) and BBC (87.4%). However, also note that the best overall performance is attained when our pre-trained Wikipedia sense embeddings are used. This highlights the effectiveness of Wikipedia in handling domain-specific entities, thanks to its broad sense inventory.

8.4.3 Polarity Detection

Polarity detection is the most popular evaluation framework for sentiment analysis (Dong et al., 2015). The task is essentially a binary classification which determines if the sentiment of a given sentence or document is negative or positive.

Dataset	Type	#docs	Avg doc size	Vocab size	Coverage	Evaluation
RTC	Snippets	438,000	23.4	128,056	81.3%	Train-Test
IMDB	Reviews	50,000	268.8	140,172	82.5%	Train-Test
PL05	Snippets	10,662	21.5	19,825	81.3%	10 cross valid.
PL04	Reviews	2,000	762.1	45,077	82.4%	10 cross valid.
Stanford	Phrases	119,783	10.0	19,400	81.6%	10 cross valid.

Table 8.3. Statistics of the polarity detection datasets.

Datasets

For the polarity detection task we used five standard evaluation datasets. Table 8.1 summarizes statistics. **PL04** (Pang and Lee, 2004) is a polarity detection dataset composed of full movie reviews. **PL05**[15] (Pang and Lee, 2005), instead, is composed of short snippets from movie reviews. **RTC** contains critic reviews from Rotten Tomatoes[16], divided into 436,000 training and 2,000 test instances. **IMDB** (Maas et al., 2011) includes 50,000 movie reviews, split evenly between training and test. Finally, we used the **Stanford** Sentiment dataset (Socher et al., 2013), which associates each review with a value that denotes its sentiment. To be consistent with the binary classification of the other datasets, we removed the neutral phrases according to the dataset's scale (between 0.4 and 0.6) and considered the reviews whose values were below 0.4 as negative and above 0.6 as positive. This resulted in a binary polarity dataset of 119,783 phrases. Unlike the previous four datasets, this dataset does not contain an even distribution of positive and negative labels.

Results

Table 8.4 lists accuracy performance of our classification model and all its variants on five polarity detection datasets. Results are generally better than those of Kim (2014), showing that the addition of the recurrent layer to the model (cf. Section 8.3) was beneficial. However, interestingly, no consistent performance gain is observed in the polarity detection task, when the model is provided with disambiguated input, particularly for datasets with relatively short reviews. We attribute this to the nature of the task. Firstly, given that words rarely happen to be ambiguous with respect to their sentiment, the semantic sense distinctions provided by the disambiguation stage do not assist the classifier in better decision making, and instead introduce data sparsity. Secondly, since the datasets mostly contain short texts, e.g., sentences or snippets, the disambiguation algorithm does not have sufficient context to make high-confidence judgements, resulting in fewer disambiguations or less reliable ones. In the following section we perform a more in-depth analysis of the impact of document size on the performance of our sense-based models.

[15] Both PL04 and PL05 were downloaded from `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

[16] `http://www.rottentomatoes.com`

Initialization	Input type		RTC	IMDB	PL05	PL04	Stanford
Random	Word		83.6	87.7	77.3	67.9	91.8
	Sense	WordNet	83.2	87.4	76.6	67.4	91.3
		Wikipedia	83.1	**88.0**	75.9†	67.1	91.0
	Supersense	WordNet	**84.4**	**88.0**	75.9	66.2	91.4†
		Wikipedia	83.1	**88.4***	75.8	**69.3***	91.0
Pre-trained	Word		85.5	88.3	80.2	72.5	93.1
	Sense	WordNet	83.4	**88.3**	79.2	69.7†	92.6
		Wikipedia	83.8	87.0†	79.2	**73.1**	92.3
	Supersense	WordNet	85.2	**88.8**	79.5	**73.8**	92.7†
		Wikipedia	84.2	87.9	78.3†	**72.6**	92.2

Table 8.4. Accuracy performance on five polarity detection datasets. Given that polarity datasets are balanced[17], we do not report F1 which would have been identical to accuracy.

8.4.4 Analysis

Document size. A detailed analysis revealed a relation between document size (the number of tokens) and performance gain of our sense-level model. We show in Figure 8.3 how these two vary for our most consistent configuration, i.e. Wikipedia supersenses, with random initialization. Interestingly, as a general trend, the performance gain increases with average document size, irrespective of the classification task. We attribute this to two main factors:

1. **Sparsity**: Splitting a word into multiple word senses can have the negative side effect that the corresponding training data for that word is distributed among multiple independent senses. This reduces the training instances per word sense, which might affect the classifier's performance, particularly when senses are semantically related (in comparison to fine-grained senses, supersenses address this issue to some extent).

2. **Disambiguation quality**: As also mentioned previously, our disambiguation algorithm requires the input text to be sufficiently large so as to create a graph with an adequate number of coherent connections to function effectively. In fact, for topic categorization, in which the documents are relatively long, our algorithm manages to disambiguate a larger proportion of words in documents with high confidence. The lower performance of graph-based disambiguation algorithms on short texts is a known issue (Moro et al., 2014; Raganato et al., 2017), the tackling of which remains an area of exploration.

Senses granularity. Our results showed that reducing fine-granularity of sense distinctions can be beneficial to both tasks, irrespective of the underlying sense

[17] Stanford is the only unbalanced dataset, but F1 results were almost identical to accuracy.

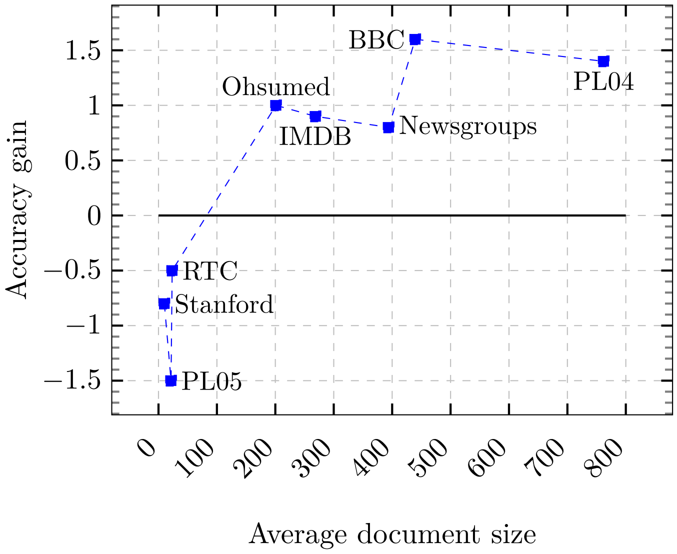


Figure 8.3. Relation between average document size and performance improvement using Wikipedia supersenses with random initialization.

inventory, i.e. WordNet or Wikipedia, which corroborates previous findings (Hovy et al., 2013; Flekova and Gurevych, 2016). This suggests that text classification does not require fine-grained semantic distinctions. In this work we used a simple technique based on WordNet's lexicographer files for coarsening senses in this sense inventory as well as in Wikipedia. We leave the exploration of this promising area as well as the use of other granularity reduction techniques for WordNet (Snow et al., 2007; Bhagwani et al., 2013) and Wikipedia (see Section 7.1) sense inventories to future work.

8.5 Conclusion

We proposed a pipeline for the integration of sense level knowledge into a state-of-the-art text classifier. We showed that a simple disambiguation of the input can lead to consistent performance gain, particularly for longer documents and when the granularity of the underlying sense inventory is reduced. Our pipeline is modular and can be used as an *in vivo* evaluation framework for WSD and word and sense representation models. This clearly differs from the word in context similarity task (Huang et al., 2012), which has been used to evaluate and compare word and sense representations in the literature. In this task WSD is also required as a first step. However, while in our framework word and sense vector representations can be directly compared using the context and evaluated on a real-word downstream NLP task, in the word in similarity task word vectors are generally evaluated in isolation and sense representations are ultimately evaluated in the more intrinsic similarity task.

We release our code and data (including pre-trained sense and supersense embeddings) at `https://github.com/pilehvar/sensecnn` to allow further checking of the choice of hyperparameters and to allow further analysis and comparison.

We hope that our work will foster future research on the integration of sense-level knowledge into downstream applications. As future work, we plan to investigate the extension of the approach to other languages and applications. Also, given the promising results observed for supersenses, we plan to investigate task-specific coarsening of sense inventories, particularly Wikipedia, or the use of SentiWordNet (Baccianella et al., 2010), which could be more suitable for polarity detection.

Chapter 9

Conclusion and Future Work

In this thesis we have investigated the construction and application of vector representations of senses, concepts and entities. These representations are aimed at solving the inherent ambiguity of language by modeling the deeper sense level. At the same time, these representations encode knowledge from lexical resources which can be valuable when integrated into current NLP architectures. We argue that sense representations are an effective middle-ground between theory and practise, as they contain theoretically-grounded semantic properties from lexical resources and are flexible to be applied in different tasks. This flexibility is similar to the versatility observed in currently ubiquitous word embeddings, but goes beyond their surface form shallow modeling.

In particular, we presented NASARI, a novel technique for the semantic representation of concepts and named entities in arbitrary languages. Our approach combines the structural knowledge from semantic networks with the statistical information derived from text corpora and Wikipedia. By exploiting the complementary knowledge of Wikipedia and WordNet, we provide effective representations for millions BabelNet synsents, including WordNet nominal synsets and a full coverage of Wikipedia concepts and entities. We also presented SW2V, an approach which is specifically targeted to capturing distributional information from large amounts of text corpora for learning word and sense embeddings in the same vector space. While NASARI leverages the vast amount of knowledge present in heterogeneous lexical resources, SW2V provides more flexibility in which it can equally learn from different corpora. Learning from different signals makes these two approaches useful for different sets of problems and compatible, as shown on the sense clustering task (Section 7.1). In this task the combination of NASARI and SW2V proved to be the most effective, as the knowledge encoded in both resources proved complementary. This opens up exciting new directions for future work, combining knowledge from semantic representations that are constructed using different signals.

We have put forward simple frameworks for making use of these representations, showing how they can be effectively applied in relevant and interconnected tasks such as Word Sense Disambiguation and downstream NLP applications like text categorization and sentiment analysis.[1] For Word Sense Disambiguation we pro-

[1] The semantic representations of NASARI have also been leveraged by other researchers using different methodologies on a set of diverse applications. For instance, they have been shown

posed a framework to integrate Nasari into a simple knowledge-based system. Our system proved highly flexible, achieving competitive results on several languages and resources. Moreover, we proposed a method which leverages comparable corpora for high-quality disambiguation and Entity Linking. This method is refined using distributional similarity based on Nasari vectors. This refinement step proved particularly reliable, contributing to a precision over 80% in most cases. Thanks to this multilingual disambiguation procedure we released two high-quality sense-annotated corpora for multiple languages: SenseDefs, consisting of textual definitions for over two hundred languages, and EuroSense, containing sense annotations for the Europarl corpus in 21 languages. The creation of these datasets may in turn help overcome the knowledge-acquisition bottleneck, as gathering a reasonable number of sense annotations for each word would require a large amount of manual effort. In fact, the integration of EuroSense as part of the training in a supervised WSD system, i.e. IMS (Zhong and Ng, 2010), was shown to provide a noticeable performance boost.

Sense representations have also proved to contribute to the improvement and enrichment of knowledge bases (Chapter 7). This creates an interesting interplay, as the improvement of knowledge bases could potentially also contribute to higher quality sense representations. We proposed a method exploiting Nasari vectors for a large-scale domain labeling of lexical resources, annotating over a million concepts and entities for resources like WordNet, Wikipedia and BabelNet, with an estimated precision over 80%. This additional domain information available in lexical resources constitute a practical tool to exploit in applications. In Section 7.3 we show how a simple clustering of the training data by domains can lead to huge improvements when integrated into a supervised hypernym discovery system. For collocation discovery we used a similar supervised model exploiting sense vector representations, obtaining encouraging results. For this task the shared space of word and sense embeddings and flexibility of our proposed SW2V model proved decisive. As future work we are planning to extend our work in hypernym and collocation discovery for learning other semantic relations as well.

In Chapter 8 we showed how sense embeddings can be seamlessly integrated into an state-of-the-art neural network text classifier. We performed an extensive evaluation on standard benchmarks of text categorization and sentiment analysis. Our analysis highlighted interesting insights which pave the way for new lines of research. For instance, our evaluation showed that improvements can be obtained when the granularity of the sense inventory is reduced. Therefore, in order to reduce this granularity, sense clustering techniques should come into play. The use of automatic and semi-automatic methods exploiting knowledge-based representations can therefore play a decisive role in this regard. Our evaluation also underlined that the sense-based pipeline yielded consistent improvements on middle-size and long documents, but its performance was less reliable on short texts (e.g. sentences). This is directly related to the WSD performance, as WSD systems, particularly

effective in diverse applications such as cross-lingual question answering (Veyseh and Pouran, 2016) common-sense knowledge representation (Lieto et al., 2016), knowledge base population (Basile et al., 2016), alignment of lexical resources (Cocos et al., 2017b), WSD (Tripodi and Pelillo, 2017) or visual object recognition (Young et al., 2016). This reinforces one of the strong points about these kinds of semantic representation, which is their versatility and flexibility.

knowledge-based, have proved to perform better when a longer context is given. On this respect, developing a highly-performing WSD system on all kinds of text should be a crucial prerequisite. Improvement in WSD is therefore fundamental, which creates an interesting symbiosis between sense representations and WSD. As future work, in addition to the methods presented in Chapter 6, we are also investigating the use of our sense embedddings for the initialization of supervised WSD and NED neural architectures, following the line of Eshel et al. (2017). This could potentially alleviate the amount of sense annotations required for producing reliable supervised WSD and EL models. Another interesting line for future work would be the integration of our sense-based pipeline with character-based neural architectures. These character-based models appear to be an interesting and effective solution to handle OOV words, especially in morphologically rich languages, while providing improvements in different NLP tasks (Ballesteros et al., 2015; Kim et al., 2016; Luong and Manning, 2016; Xiao and Cho, 2016).

Finally, we advocate for further research on developing better evaluation procedures for word and sense representations. While intrinsic evaluation of these models enable us to have a neat overview of some important properties of these models, they have been shown to often not correlate with downstream task performance (Tsvetkov et al., 2015; Chiu et al., 2016). Therefore, a complementary evaluation on downstream tasks would be desirable. Our evaluation framework on text categorization and sentiment analysis represents a first step towards this goal. In our framework both word and sense embeddings can be directly plugged in and used for initialization in a state-of-the-art neural network architecture. This enables a direct comparison of word and sense representations on two real-world NLP tasks. Additionally, as a more direct way of comparing representations, we proposed the *outlier detection* task (Camacho-Collados and Navigli, 2016). This task, inspired by conventional vocabulary questions in language tests (Richards, 1976), consists of finding the word that does not belong in a given set of words[2]. We proposed an evaluation framework for this task which is aimed at evaluating vector representations, capturing interesting semantic properties of the vector space. This evaluation framework, based on a clear and well-defined gold standard[3], constitutes an interesting middle-ground between intrinsic and extrinsic evaluation of vector space models.[4] As far as strictly intrinsic evaluation is concerned, we proposed an extension of current English word similarity datasets to other languages, additionally enabling a direct comparison across languages (see Section 4.3.3). Building upon this idea, we recently presented a SemEval shared task on multilingual and cross-lingual semantic word similarity for five languages. This evaluation framework consists of high-quality monolingual and cross-lingual datasets which are aimed at solving some of the deficiencies of previous evaluation datasets: these datasets are composed of a balanced set of concepts and named entities (including multiword expressions)

[2] For example, given the set of words *apple*, *banana*, *lemon*, *book* and *orange*, *book* would be an outlier as it is not a fruit like the others.

[3] Batchkarov et al. (2016) criticized word similarity datasets for their relatively low inter-annotator agreement, which prevented from drawing reliable conclusions. Instead, the outlier detection task, as shown as part of its validation, consists of well-defined gold standard, with IAA figures over 98%.

[4] Blair et al. (2017) extended our work by proposing a method for constructing multilingual outlier detection datasets automatically.

from a wide variety of domains, annotated by experts with a high inter-annotator agreement and with a clear distinction between similarity and relatedness. We hope our efforts in this area will contribute to the development of reliable evaluation frameworks for word and sense vector space models, and foster further research on building better intrinsic and extrinsic evaluation procedures.

List of released resources

As a result of all the work of this thesis, we release to the research community various data resources and open code. In this thesis we believe in the importance of reproducibility and accessibility in research. That is the reason why we have made available most of the resources obtained during and as a result of our work, in the hope that they are useful for the community but also contribute to the reproducibility of our experiments.

NASARI

We release lexical and unified vectors of Nasari (Section 4) for five languages (English, French, German, Italian and Spanish) at `lcl.uniroma1.it/nasari`. Additionally, we release various versions of the embedded Nasari vectors for English and Spanish, sharing the same vector space of word embedddings.

SW2V

For SW2V (Section 5) we release the code for jointly training word and sense embeddings and various pre-trained models. Preprocessed corpora using our shallow word-sense connectivity algorithm are also available. Data and code are available at `lcl.uniroma1.it/sw2v`.

Multilingual Word Similarity Datasets

The monolingual and cross-lingual word similarity datasets constructed as part of the intrinsic evaluation (Section 4.3) are made available from the following website: `lcl.uniroma1.it/similarity-datasets`. The languages available are for English, Farsi, French, German, Portuguese and Spanish, and all their pairwise combinations. These datasets are constructed taking the RG-65 dataset (Rubenstein and Goodenough, 1965) as reference, including an alignment-based algorithm for constructing the cross-lingual datasets (Camacho-Collados et al., 2015a). The code for constructing cross-lingual similarity datasets from aligned monolingual datasets is also available online.

SenseDefs

We release SenseDefs (Section 6.3.2), a sense-annotated corpus of textual definitions featuring over 38 million definitions in 263 languages. We release two versions of the corpus: full (high-coverage) and refined (high-precision). It is available for download at `lcl.uniroma1.it/sensedefs` and is available in two different formats: a human- and machine-readable XML divided by language and resource, and NIF.

- *XML format.* The format for each of the two versions of SenseDefs (*full* and *refined*) is almost identical: the corpus is first divided by resource (WordNet, Wikipedia, Wiktionary, Wikidata and OmegaWiki) and then divided by language within each resource. Finally, the disambiguated glosses for each language and resource are stored in standard XML files.

- *NIF format.* Recently the Linked Open Data community has made considerable efforts to extract and standardize structured knowledge from a wide range of corpora and linguistic resources, making them available on the Web by means of the RDF format (Chiarcos et al., 2011; Auer and Hellmann, 2012; Ehrmann et al., 2014; Flati and Navigli, 2014). In order to simplify the interoperability of linguistic resources, the NLP Interchange Format (NIF) was developed (Hellmann et al., 2013). NIF aims at easing the use of Linked Data among Natural Language Processing tools, language resources and annotations. Following this overarching goal, several resources have already been converted and made available on NIF format, contributing to the creation of the Linguistic Linked Open Data (Rizzo et al., 2012; Hellmann et al., 2012; Röder et al., 2014). We transformed the English annotations of the refined version of SenseDefs into the NLP Interchange Format, following the guidelines provided by the hackathon organized at the Multilingual Linked Open Data for Enterprises Workshop (MLODE 2014)[5].

EuroSense

EuroSense (Section 6.3.3) is available at `lcl.uniroma1.it/eurosense`. We release two different versions of the corpus, using the same XML format as SenseDefs:

- A *high-coverage version*, obtained after the first stage of the pipeline, i.e. multilingual joint disambiguation with Babelfy. Here, each sense annotation is associated with a coherence score;

- A *high-precision version*, obtained after the similarity-based refinement with Nasari. In this version, sense annotations are associated with both a coherence score and a distributional similarity score.

BabelDomains

BabelDomains, a unified resource including domain labels for BabelNet, Wikipedia and WordNet (Section 7.2.3), is available for download at `lcl.uniroma1.it/`

[5] `http://wwwusers.di.uniroma1.it/~flati/hackathon/index.html`

babeldomains. In the release we include a confidence score[6] for each domain label. Additionally, the domain labels have been integrated into BabelNet[7], both in the API and in the online interface[8].

TaxoEmbed

Code and data for TaxoEmbed (Section 7.3) are available at http://wwwusers.di.uniroma1.it/~dellibovi/taxoembed/.[9] The code consists of simple Python scripts for making use of TaxoEmbed. The data include the Wikidata pairs used for both training and testing. Both the domain clusters used in the evaluation and the domain labels of BabelDomains are available.

ColWordNet

We release ColWordNet (CWM, Section 7.4) at several different confidence levels. The version with the highest confidence includes over 100k collocational edges, which connect over 8k unique base and collocate WordNet synsets. These connections are further enriched by two pieces of information, namely (1) the type of collocation (e.g. 'intense' or 'perform'), and (2) a confidence score derived from our approach. Moreover, in addition to CWN, we also release four modified versions of the well-known Word2Vec Google News vector space model, retrofitted with collocational information, which we constructed for the extrinsic evaluation of CWN. These models can be exploited both for assessing the correctness of a collocation and for the discovery of alternative collocates for a given collocation. Finally, we also make available the evaluation datasets built as part of the Collocational Sensitivity experiment. All data associated with this publication is publicly available at https://bitbucket.org/luisespinosa/cwn/.

SenseCNN

Code and data for the experiments on text categorization and sentiment analysis (Section 8) are available at https://github.com/pilehvar/sensecnn. Data includes pre-trained word and sense embeddings used in the experiments as well as the supersense clustering files for both WordNet and Wikipedia.

[6] The confidence score for each synset's domain label is computed as the relative number of neighbours in the BabelNet semantic network sharing the same domain.

[7] In its current 3.7 release version we have included two additional domains to the ones included in Table 7.2: Farming and Textile and Clothing

[8] See http://babelnet.org/search?word=house&lang=EN for an example of the domain annotations of all senses of *house* in BabelNet.

[9] Code is also directly available at https://bitbucket.org/luisespinosa/taxoembed

List of Acronyms

- AI: Artificial Intelligence
- BOW: Bag of Words
- EL: Entity Linking
- IAA: Inter-Annotator Agreement
- KB: Knowledge Base
- MWE: MultiWord Expressions
- NED: Named Entity Disambiguation
- NLP: Natural Language Processing
- NN: Nearest Neighbours
- OIE: Open Information Extraction
- OOV: Out-Of-Vocabulary
- PoS: Part-of-Speech
- SVM: Support Vector Machine
- VSM: Vector Space Model
- WSD: Word Sense Disambiguation

Bibliography

Eneko Agirre and Oier Lopez de Lacalle. Publicly available topic signatures for all WordNet nominal senses. In *Proceedings of LREC*, pages 1123–1126, Lisbon, Portugal, 2004.

Eneko Agirre and Oier Lopez. Clustering WordNet word senses. In *Proceedings of Recent Advances in Natural Language Processing*, pages 121–130, Borovets, Bulgaria, 2003.

Eneko Agirre and Aitor Soroa. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of EACL*, pages 33–41, 2009.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of NAACL*, pages 19–27, 2009a.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1501–1506, Pasadena, California, 2009b.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84, 2014.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, 2013.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning Principled Bilingual Mappings of Word Embeddings while Preserving Monolingual Invariance. In *Proc. of EMNLP*, pages 2289–2294, 2016.

Sören Auer and Sebastian Hellmann. The web of data: Decentralized, collaborative, interlinked and interoperable. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*, 2012.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

Miguel Ballesteros, Chris Dyer, and Noah A Smith. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of EMNLP*, 2015.

Satanjeev Banerjee and Ted Pedersen. Extended gloss overlap as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, Mexico, 2003.

Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. Structured learning for taxonomy induction with belief propagation. In *Proceedings of ACL*, pages 1041–1051, 2014.

Daniel Bär, Torsten Zesch, and Iryna Gurevych. DKPro similarity: An open source framework for text similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 121–126, Sofia, Bulgaria, August 2013.

Marco Baroni and Alessandro Lenci. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics, 2011.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of EACL*, pages 23–32, 2012.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, pages 238–247, 2014.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland, 2014.

Valerio Basile, Soufian Jebbara, Elena Cabrio, and Philipp Cimiano. Populating a knowledge base with object-location relations using distributional semantics. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 34–50. Springer, 2016.

Osman Başkaya and David Jurgens. Semi-supervised learning with induced word senses for state of the art word sense disambiguation. *Journal of Artificial Intelligence Research*, 55:1025–1058, 2016.

Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. A critique of word similarity as a method for evaluating distributional semantic models. In *ACL Workshop on Evaluating Vector Space Representations for NLP*, Berlin, Germany, 2016.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3: 1137–1155, 2003.

Andrew Bennett, Timothy Baldwin, Jey Han Lau, Diana McCarthy, and Francis Bond. Lexsemtm: A semantic dataset based on all-words unsupervised sense distribution learning. In *Proceedings of ACL*, pages 1513–1524, 2016.

Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Proceedings of the Workshop on Multilingual Linguistic Ressources*, pages 101–108. Association for Computational Linguistics, 2004.

Sumit Bhagwani, Shrutiranjan Satapathy, and Harish Karnick. Merging word senses. In *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, pages 11–19, Seattle, Washington, USA, October 2013.

Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. Finding the right facts in the crowd: factoid question answering over social media. In *Proceedings of the 17th international conference on World Wide Web*, pages 467–476. ACM, 2008.

Mokhtar-Boumeyden Billami, José Camacho-Collados, Evelyne Jacquey, and Laurence Kister. Annotation sémantique et validation terminologique en texte intégral en SHS. In *Proceedings of TALN*, pages 363–376, 2014.

Philip Blair, Yuval Merhav, and Joel Barry. Automated generation of multilingual clusters for the evaluation of distributed representations. In *Proceedings of ICLR: Workshop track*, 2017.

Guido Boella and Luigi Di Caro. Supervised learning of syntactic contexts for uncovering definitions and extracting hypernym relations in text databases. In *Machine learning and knowledge discovery in databases*, pages 64–79. Springer, 2013.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the SemEval workshop*, 2015.

Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In *Proceedings of EMNLP*, pages 615–620, 2014.

G. Bouma. Collocation Extraction beyond the Independence Assumption. In *Proceedings of ACL, Short papers*, pages 109–114, Uppsala, Sweden, 2010.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(1-47), 2014.

Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

Hiram Calvo and Alexander Gelbukh. Is the most frequent sense of a word better connected in a semantic network? In *International Conference on Intelligent Computing*, pages 491–499. Springer, 2015.

José Camacho-Collados and Roberto Navigli. Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. In *ACL Workshop on Evaluating Vector Space Representations for NLP*, pages 43–50, Berlin, Germany, 2016.

Jose Camacho-Collados and Roberto Navigli. BabelDomains: Large-Scale Domain Labeling of Lexical Resources. In *Proceedings of EACL (2)*, pages 223–228, Valencia, Spain, 2017.

José Camacho-Collados, Mokhtar Billami, Evelyne Jacquey, and Laurence Kister. Approche statistique pour le filtrage terminologique des occurrences de candidats termes en texte intégral. In *Proceedings of JADT*, pages 121–133, 2014.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. A Framework for the Construction of Monolingual and Cross-lingual Word Similarity Datasets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing – Short Papers*, pages 1–7, Beijing, China, 2015a.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. NASARI: a Novel Approach to a Semantically-Aware Representation of Items. In *Proceedings of NAACL*, pages 567–577, 2015b.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. A Unified Multilingual Semantic Representation of Concepts. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 741–751, Beijing, China, 2015c.

José Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato, and Roberto Navigli. A Large-Scale Multilingual Disambiguation of Glosses. In *Proceedings of LREC*, pages 1701–1708, Portoroz, Slovenia, 2016a.

José Camacho-Collados, Ignacio Iacobacci, Roberto Navigli, and Mohammad Taher Pilehvar. Semantic representations of word senses and concepts. In *ACL Tutorial*, 2016b.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016c.

Cristian Cardellino. Spanish Billion Words Corpus and Embeddings, March 2016. URL `http://crscardellino.me/SBWCE/`.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of AAAI*, pages 1306–1313, 2010.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP*, pages 1025–1035, Doha, Qatar, 2014.

Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of EMNLP*, pages 1787–1796, Seattle, Washington, 2013.

Christian Chiarcos, Sebastian Hellmann, and Sebastian Nordhoff. Towards a linguistic linked open data cloud: The open linguistics working group. *TAL*, 52(3):245–275, 2011.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the Workshop on Evaluating Vector Space Representations for NLP, ACL*, 2016.

François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

Y. Choueka. Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. In *Proceedings of the RIAO*, pages 34–38, 1988.

K. Church and P. Hanks. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of ACL*, pages 76–83, 1989.

Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. Word sense filtering improves embedding-based lexical substitution. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 110–119, Valencia, Spain, 2017a.

Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. Mapping the paraphrase database to wordnet. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 84–90, 2017b.

J. A. Cohen. A coefficient of agreement of nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011. ISSN 1532-4435.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. Trans-gram, Fast Cross-lingual Word-embeddings. In *Proc. of EMNLP*, pages 1109–1113, 2015.

A. Cowie. Phraseology. In R.E. Asher and J.M.Y. Simpson, editors, *The Encyclopedia of Language and Linguistics, Vol. 6*, pages 3168–3171. Pergamon, Oxford, 1994.

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9(Aug):1757–1774, 2008.

Silviu Cucerzan. Large-scale Named Entity Disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, 2007.

Bharath Dandala, Chris Hokamp, Rada Mihalcea, and Razvan C. Bunescu. Sense clustering using Wikipedia. In *Proceedings of Recent Advances in Natural Language Processing*, pages 164–171, Hissar, Bulgaria, 2013a.

Bharath Dandala, Rada Mihalcea, and Razvan Bunescu. Word sense disambiguation using wikipedia. In *The PeopleâĂŹs Web Meets NLP*, pages 241–262. Springer, 2013b.

Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of ACL*, pages 256–263, Prague, Czech Republic, 2007.

Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

Claudio Delli Bovi, Luis Espinosa Anke, and Roberto Navigli. Knowledge base unification via sense embeddings and disambiguation. In *Proceedings of EMNLP*, pages 726–736, Lisbon, Portugal, 2015a.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543, 2015b. ISSN 2307-387X.

Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. Eurosense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of ACL (2)*, pages 594–600, Vancouver, Canada, 2017.

Li Dong, Furu Wei, Shujie Liu, Ming Zhou, and Ke Xu. A statistical parsing framework for sentiment classification. *Computational Linguistics*, 41(2):293–336, June 2015. ISSN 0891-2017.

Patrick Drouin. Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115, 2003.

Philip Edmonds and Scott Cotton. Senseval-2: Overview. In *Proceedings of The Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–6, Toulouse, France, 2001.

Maud Ehrmann, Francesco Cecconi, Daniele Vannella, John Philip McCrae, Philipp Cimiano, and Roberto Navigli. Representing multilingual data as linked data: the case of babelnet 2.0. In *LREC*, pages 401–408, 2014.

Katrin Erk. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic*, pages 216–223, 2007.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554, 2013.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. Named entity disambiguation for noisy text. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 58–68, Vancouver, Canada, 2017. URL `http://aclweb.org/anthology/K17-1008`.

Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of EMNLP*, pages 424–435, Austin, Texas, 2016a.

Luis Espinosa-Anke, Jose Camacho-Collados, Sara Rodrıguez-Fernández, Horacio Saggion, and Leo Wanner. Extending wordnet with fine-grained collocational information via supervised distributional learning. In *Proceedings of COLING*, pages 3422–3432, Osaka, Japan, 2016b.

Luis Espinosa-Anke, Horacio Saggion, Francesco Ronzano, and Roberto Navigli. Extasem! extending, taxonomizing and semantifying domain terminologies. In *Proceedings of AAAI*, Phoenix, USA, 2016c.

Allyson Ettinger, Philip Resnik, and Marine Carpuat. Retrofitting Sense-Specific Word Vectors Using Parallel Text. In *Proceedings of NAACL-HLT*, pages 1378–1383, 2016.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165 (1):91–134, 2005.

S. Evert. Corpora and Collocations. In A. Lüdeling and M. Kytö, editors, *Corpus Linguistics. An International Handbook*. Mouton de Gruyter, Berlin, 2007.

Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of CoNLL*, pages 260–269, 2016.

Stefano Faralli and Roberto Navigli. A New Minimally-supervised Framework for Domain Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1411–1422, Jeju, Korea, 2012.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, pages 1606–1615, 2015.

Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.

Lev Finkelstein, Gabrilovich Evgenly, Matias Yossi, Rivlin Ehud, Solan Zach, Wolfman Gadi, and Ruppin Eytan. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, 2002. ISSN 1046-8188.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

Tiziano Flati and Roberto Navigli. Three birds (in the llod cloud) with one stone: Babelnet, babelfy and the wikipedia bitaxonomy. In *Proc. of SEMANTiCS2014*, 2014.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Two is bigger (and better) than one: the Wikipedia Bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 945–955, Baltimore, USA, 2014.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artificial Intelligence*, 241:66–102, 2016.

Lucie Flekova and Iryna Gurevych. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of ACL*, pages 2029–2041, 2016.

Trevor Fountain and Mirella Lapata. Taxonomy induction using hierarchical random graphs. In *Proceedings of NAACL*, pages 466–476. Association for Computational Linguistics, 2012.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning semantic hierarchies via word embeddings. In *Proceedings of ACL*, pages 1199–1209, Baltimore, USA, 2014.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, pages 1606–1611, 2007.

William A. Gale, Kenneth Church, and David Yarowsky. A method for disambiguating word senses in a corpus. *Computers and the Humanities*, 26:415–439, 1992.

Z.M. Gao. Automatic Identification of English Collocation Errors based on Dependency Relations. *Sponsors: National Science Council, Executive Yuan, ROC Institute of Linguistics, Academia Sinica NCCU Office of Research and Development*, page 550, 2013.

Peter Gärdenfors. *Conceptual spaces: The geometry of thought.* MIT press, 2004.

A. Gelbukh and O. Kolesnikova. *Semantic Analysis of Verbal Collocations with Lexical Functions.* Springer, Heidelberg, 2012.

Oren Glickman, Ido Dagan, and Moshe Koppel. A probabilistic classification approach for lexical textual entailment. In *Proceedings of the National Conference On Artificial Intelligence*, page 1050, 2005.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520, Bellevue, Washington, USA, 2011.

Josu Goikoetxea, Aitor Soroa, Eneko Agirre, and Basque Country Donostia. Random walks and neural network language models on knowledge bases. In *Proceedings of NAACL*, pages 1434–1439, 2015.

Hila Gonen and Yoav Goldberg. Semi Supervised Preposition-Sense Disambiguation using Multilingual Data. In *Proc. of COLING*, pages 2718–2729, 2016.

Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *Computer Vision–ECCV 2014*, pages 529–545. Springer, 2014.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. BilBOWA: Fast Bilingual Distributed Representations without Word Alignments. In *Proc. of ICML*, pages 748–756, 2015.

Roger Granada, Cassia Trojahn, and Renata Vieira. Comparing semantic relatedness between word pairs in Portuguese using Wikipedia. In *Computational Processing of the Portuguese Language*, pages 170–175. 2014.

Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International conference on Machine learning*, pages 377–384. ACM, 2006.

Adam Grycner and Gerhard Weikum. Harpy: Hypernyms and alignment of relational paraphrases. In *Proceedings of COLING*, pages 2195–2204, Dublin, Ireland, 2014.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of COLING*, pages 497–507, 2014.

Weiwei Guo and Mona T. Diab. Combining orthogonal monolingual and multilingual sources of evidence for all words WSD. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1542–1551, Uppsala, Sweden, 2010.

Iryna Gurevych. Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of IJCNLP*, pages 767–778. 2005.

Yoan Gutiérrez, Yenier Castañeda, Andy González, Rainel Estrada, D. Dennys Piug, I. Jose Abreu, Roger Pérez, Antonio Fernández Orquín, Andrés Montoyo, Rafael Muñoz, and Franc Camara. UMCC_DLSI: Reinforcing a ranking algorithm with sense frequencies and multidimensional semantic resources to solve multilingual word sense disambiguation. In *Proceedings of SemEval 2013*, pages 241–249, 2013.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of EMNLP*, pages 595–605, Austin, Texas, 2016.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. UMBC EBIQUITY-CORE: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 44–52, 2013.

Sanda M Harabagiu, Steven J Maiorano, and Marius A Pasca. Open-domain textual question answering techniques. *Natural Language Engineering*, 9(03):231–267, 2003.

Zellig Harris. Distributional structure. *Word*, 10:146–162, 1954.

Samer Hassan and Rada Mihalcea. Semantic relatedness using salient semantic analysis. In *Proceedings of AAAI*, pages 884–889, 2011.

Taher H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, pages 517–526, Hawaii, USA, 2002.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING 1992*, pages 539–545, 1992.

Serge Heiden, Jean-Philippe Magué, Bénédicte Pincemin, et al. Txm: Une plateforme logicielle open-source pour la textométrie-conception et développement. In *Statistical Analysis of Textual Data-Proceedings of 10th International Conference Journées d'Analyse statistique des Données Textuelles*, volume 2, pages 1021–1032, Rome, Italy, 2010.

Sebastian Hellmann, Claus Stadler, and Jens Lehmann. The german dbpedia: A sense repository for linking entities. In *Linked Data in Linguistics*, pages 181–190. Springer, 2012.

Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating nlp using linked data. In *International Semantic Web Conference*, pages 98–113. Springer, 2013.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SemEval: Recent Achievements and Future Directions*, pages 94–99, 2009.

Karl Moritz Hermann and Phil Blunsom. Multilingual Distributed Representations without Word Alignment. In *Proc. of ICLR*, pages 1–9, 2014.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2015.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of CIKM*, pages 545–554, 2012.

Johannes Hoffart, Dragan Milchevski, and Gerhard Weikum. Stics: searching with strings, things, and cats. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1247–1248. ACM, 2014.

Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1):177–196, 2001.

Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27, 2013.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*, pages 873–882, Jeju Island, South Korea, 2012.

Sung Ju Hwang, Kristen Grauman, and Fei Sha. Semantic kernel forests from multiple taxonomies. In *Advances in Neural Information Processing Systems*, pages 1718–1726, 2012.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*, pages 95–105, Beijing, China, 2015.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of ACL*, pages 897–907, 2016.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of NAACL*, pages 683âĂŞ–693, 2015.

Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of ACL*, pages 264–271, Prague, Czech Republic, 2007.

Peng Jin, Diana McCarthy, Rob Koeling, and John Carroll. Estimating and exploiting the entropy of sense distributions. In *Proceedings of NAACL (2)*, pages 233–236, 2009.

Richard Johansson and Luis Nieto Pina. Embedding a semantic network in a word space. In *Proceedings of NAACL*, pages 1428–1433, 2015.

Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of NAACL*, pages 103–112, Denver, Colorado, 2015.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

Michael P. Jones and James H. Martin. Contextual spelling correction using latent semantic analysis. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, pages 166–173, 1997.

Colette Joubarne and Diana Inkpen. Comparison of semantic similarity for different languages using the Google n-gram corpus and second-order co-occurrence measures. In *Advances in Artificial Intelligence*, pages 216–221. 2011.

David Jurgens and Mohammad Taher Pilehvar. Reserating the awesometastic: An automatic extension of the wordnet taxonomy for novel terms. In *Proceedings of NAACL*, pages 1459–1465, 2015.

David Jurgens and Mohammad Taher Pilehvar. SemEval-2016 Task 14: Semantic taxonomy enrichment. In *Proceedings of SemEval*, pages 1092–1102, 2016.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. Semeval-2014 task 3: Cross-level semantic similarity. *SemEval 2014*, pages 17–26, 2014.

Mikael Kågebäck and Hans Salomonsson. Word sense disambiguation using a bidirectional lstm. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 51–56, 2016.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665, Baltimore, USA, 2014.

Abhay Kashyap, Lushan Han, Roberto Yus, Jennifer Sleeman, Taneeya Satyapanich, Sunil Gandhi, and Tim Finin. Meerkat mafia: Multilingual and cross-level semantic textual similarity systems. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 416–423. Association for Computational Linguistics, 2014.

Alistair Kennedy and Graeme Hirst. Measuring semantic relatedness across languages. In *Proceedings of xLiTe: Cross-Lingual Technologies Workshop at the Neural Information Processing Systems Conference*, 2012.

L. Y. Keok and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the* 7^{th} *Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Philadelphia, USA, 2002.

Adam Kilgariff. Collocationality (and How to Measure it). In *Proceedings of the Euralex Conference*, pages 997–1004, Turin, Italy, 2006. Springer-Verlag.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751, Doha, Qatar, 2014.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Proceedings of AAAI*, pages 2741–2749, Phoenix, Arizona, 2016.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT summit*, volume 5, pages 79–86, 2005.

Zornitsa Kozareva and Eduard Hovy. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of EMNLP*, pages 1110–1118, 2010.

Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran S. da Silva, and Juliana S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Rec.*, 31(2):84–93, 2002.

Pierre Lafon. Sur la variabilité de la fréquence des formes dans un corpus. *Mots*, 1: 127–165, 1980.

Thomas K. Landauer and Susan T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

Tom Landauer and Scott Dooley. Latent semantic analysis: theory, method and application. In *Proceedings of CSCL*, pages 742–743, 2002.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, Tahoe City, California, 1995.

Ludovic Lebart, A Salem, and Lisette Berry. *Exploring textual data.* Kluwer Academic Publishers, 1998. ISBN 0792348400.

Dik L Lee, Huei Chuang, and Kent Seamons. Document ranking and the vector-space model. *IEEE software*, 14(2):67–75, 1997.

Els Lefever and Veronique Hoste. SemEval-2010 task 3: Cross-lingual Word Sense Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 15–20, 2010.

Els Lefever and Veronique Hoste. SemEval-2013 Task 10: Cross-lingual Word Sense Disambiguation. In *Proceedings of SemEval 2013*, pages 158–166, Atlanta, USA, 2013.

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual Conference on Systems Documentation,* Toronto, Ontario, Canada, pages 24–26, 1986.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015a.

Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. Do supervised distributional methods really learn lexical inference relations? In *NAACL 2015*, Denver, Colorado, USA, 2015b.

Jiwei Li and Dan Jurafsky. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1722âĂŞ–1732, Lisbon, Portugal, 2015.

Antonio Lieto, Enrico Mensa, and Daniele P Radicioni. A resource-driven approach for anchoring linguistic resources to conceptual spaces. In *AI* IA 2016 Advances in Artificial Intelligence*, pages 435–449. Springer, 2016.

Nut Limsopatham and Nigel Collier. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of ACL*, pages 1014–1023, Berlin, Germany, 2016.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, 1998.

Xiao Ling, Sameer Singh, and Daniel S Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015.

Frederick Liu, Han Lu, and Graham Neubig. Handling homographs in neural machine translation. *arXiv preprint arXiv:1708.06510*, 2017.

Maddalen Lopez de la Calle, Itziar Aldabe, Egoitz Laparra, and German Rigau. Predicate Matrix. Atomatically extending the semantic interoperability between predicate resources. *Language Resources and Evaluation*, 50(2):263–289, 2016.

Wei Lu, Hai Leong Chieu, and Jonathan Löfgren. A general regularization framework for domain adaptation. In *Proceedings of EMNLP*, pages 950–954, Austin, Texas, 2016.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28 (2):203–208, 1996.

Minh-Thang Luong and Christopher D. Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Association for Computational Linguistics (ACL)*, Berlin, Germany, 2016.

Tuan Luu Anh, Jung-jae Kim, and See Kiong Ng. Taxonomy construction using syntactic contextual evidence. In *Proceedings of EMNLP*, pages 810–819, 2014.

Tuan Luu Anh, Jung-jae Kim, and See-Kiong Ng. Incorporating trustiness and collective synonym/contrastive evidence into taxonomy construction. In *Proceedings of EMNLP*, pages 1013–1022, 2015.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of ACL-HLT*, pages 142–150, Portland, Oregon, USA, 2011.

Bernardo Magnini and Gabriela Cavaglià. Integrating subject field codes into WordNet. In *Proceedings of LREC*, pages 1413–1418, Athens, Greece, 2000.

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(04):359–373, 2002.

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of CONLL*, Vancouver, Canada, 2017.

Steve L. Manion. Sudoku: Treating word sense disambiguation & entity linking as a deterministic problem–via an unsupervised & iterative approach. *9th International Workshop on Semantic Evaluation (SemEval 2015)*, page 365, 2015.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

Ernst Mayr. *The growth of biological thought: Diversity, evolution, and inheritance.* Harvard University Press, 1982.

Diana McCarthy. Relating WordNet senses for word sense disambiguation. In *Proceedings of the Workshop on Making Sense of Sense at EACL-06*, pages 17–24, Trento, Italy, 2006.

Diana McCarthy and Roberto Navigli. The English lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159, 2009.

Diana McCarthy, Marianna Apidianaki, and Katrin Erk. Word sense clustering and clusterability. *Computational Linguistics*, 2016.

Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proceedings of CONLL*, pages 51–61, 2016.

I.A. Mel'čuk. Lexical Functions: A tool for the description of lexical relations in the lexicon. In L. Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–102. Benjamins Academic Publishers, Amsterdam, 1996.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

Rada Mihalcea. Using Wikipedia for automatic Word Sense Disambiguation. In *Proc. of NAACL-HLT-07*, pages 196–203, Rochester, NY, 2007.

Rada Mihalcea and Andras Csomai. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge management*, pages 233–242, Lisbon, Portugal, 2007.

Rada Mihalcea and Dan Moldovan. An automatic method for generating sense tagged corpora. In *Proceedings AAAI '99*, pages 461–466, Orlando, Florida, USA, 1999.

Rada Mihalcea and Dan Moldovan. Automatic generation of a coarse grained WordNet. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, Pittsburgh, USA, 2001.

Rada Mihalcea and Janyce Wiebe. Simcompass: Using deep learning word embeddings to assess cross-level similarity. *SemEval 2014*, page 560, 2014.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a. URL `http://arxiv.org/abs/1301.3781`.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013d.

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

George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4): 235–244, 1990.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, N.J., 1993.

David Milne and Ian H. Witten. Learning to link with Wikipedia. In *Proc. of CIKM-08*, pages 509–518, 2008.

Pol Moreno, Gabriela Ferraro, and Leo Wanner. Can we determine the semantics of collocations without using semantics? In I. Kosem, J. Kallas, P. Gantar, S. Krek, M. Langemets, and M. Tuulik, editors, *Proceedings of the eLex 2013 conference*, Tallinn & Ljubljana, 2013. Trojina, Institute for Applied Slovene Studies & Eesti Keele Instituut.

Andrea Moro and Roberto Navigli. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*, pages 288–297, 2015.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244, 2014.

Nikola Mrkšic, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica GaÅąiÄĞ, Anna Korhonen, and Steve Young. Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *TACL*, 2017.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proceedings of EMNLP-CoNLL*, pages 1135–1145, 2012.

Roberto Navigli. Meaningful clustering of senses helps boost Word Sense Disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 105–112, Sydney, Australia, 2006.

Roberto Navigli. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69, 2009.

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

Roberto Navigli and Paola Velardi. Learning word-class lattices for definition and hypernym extraction. In *ACL*, pages 1318–1327, 2010.

Roberto Navigli, Stefano Faralli, Aitor Soroa, Oier de Lacalle, and Eneko Agirre. Two birds with one stone: Learning semantic models for text categorization and Word Sense Disambiguation. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM)*, pages 2317–2320, Glasgow, UK, 2011.

Roberto Navigli, David Jurgens, and Daniele Vannella. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of SemEval 2013*, pages 222–231, 2013.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of EMNLP*, pages 1059–1069, Doha, Qatar, 2014.

J. H. Neely, D. E. Keefe, and K. L. Ross. Semantic priming in the lexical decision task: Roles of prospective prime-generated expectancies and retrospective semantic matching. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, pages 1003–1019, 1989.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of ACL*, pages 454–459, 2016.

Elisabeth Niemann and Iryna Gurevych. The people's web meets linguistic knowledge: automatic sense alignment of Wikipedia and WordNet. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 205–214, 2011.

Arantxa Otegi, Nora Aranberri, Antonio Branco, Jan Hajic, Steven Neale, Petya Osenova, Rita Pereira, Martin Popel, Joao Silva, Kiril Simov, and Eneko Agirre. QTLeap WSD/NED Corpora: Semantic Annotation of Parallel Corpora in Six Languages. In *Proc. of LREC*, pages 3023–3030, 2016.

Martha Palmer, Hoa Dang, and Christiane Fellbaum. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163, 2007.

Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone Paolo Ponzetto, and Chris Biemann. Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation. In *Proceedings of EACL*, pages 86–98, 2017.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, pages 51–61, Barcelona, Spain, 2004.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124, Ann Arbor, Michigan, 2005.

Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of KDD*, pages 613–619, 2002.

Rebecca J. Passonneau, Collin Baker, Christiane Fellbaum, and Nancy Ide. The MASC Word Sense Sentence Corpus. In *Proc. of LREC*, pages 3025–3030, 2012.

Pavel Pecina. A Machine Learning Approach to Multiword Expression Extraction. In *Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 54–57, Marrakech, 2008.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, 2014.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of ACL*, pages 21–26, 2015.

Mohammad Taher Pilehvar and Nigel Collier. De-conflated semantic representations. In *Proceedings of EMNLP*, Austin, TX, 2016.

Mohammad Taher Pilehvar and Roberto Navigli. A robust approach to aligning heterogeneous lexical resources. In *Proceedings of ACL*, pages 468–478, 2014a.

Mohammad Taher Pilehvar and Roberto Navigli. A large-scale pseudoword-based evaluation framework for state-of-the-art Word Sense Disambiguation. *Computational Linguistics*, 40(4):837–881, 2014b.

Mohammad Taher Pilehvar and Roberto Navigli. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128, 2015.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align, Disambiguate and Walk: a Unified Approach for Measuring Semantic Similarity. In *Proceedings of ACL*, pages 1341–1351, 2013.

Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. Towards a Seamless Integration of Word Senses into Downstream NLP Applications. In *Proceedings of ACL*, Vancouver, Canada, 2017.

Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich Word Sense Disambiguation rivaling supervised system. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1522–1531, Uppsala, Sweden, 2010.

Simone Paolo Ponzetto and Michael Strube. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research (JAIR)*, 30:181–212, 2007.

Simone Paolo Ponzetto and Michael Strube. Wikitaxonomy: A large scale knowledge resource. In *Proceedings of ECAI*, volume 178, pages 751–752, 2008.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*, pages 87–92, 2007.

John Prager, Jennifer Chu-Carroll, Eric W Brown, and Krzysztof Czuba. Question answering by predictive annotation. In *Advances in Open Domain Question Answering*, pages 307–347. Springer, 2008.

Thomas Proisl, Stefan Evert, Paul Greiner, and Besim Kabashi. Semantiklue: Robust semantic similarity at multiple levels using maximum weight matching. *SemEval 2014*, pages 532–540, 2014.

Lin Qiu, Kewei Tu, and Yong Yu. Context-dependent sense embedding. In *Proceedings of EMNLP*, pages 183–191, Austin, Texas, 2016.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 337–346, 2011.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. Automatic Construction and Evaluation of a Large Semantically Enriched Wikipedia. In *Proceedings of IJCAI*, pages 2894–2900, New York City, NY, USA, July 2016.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proceedings of EACL*, pages 99–110, 2017.

Joseph Reisinger and Raymond J. Mooney. Multi-prototype vector-space models of word meaning. In *Proceedings of ACL*, pages 109–117, 2010.

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*, pages 448–453, 1995.

Jack C Richards. The role of vocabulary teaching. *TESOl Quarterly*, pages 77–89, 1976.

Giuseppe Rizzo, Raphaël Troncy, Sebastian Hellmann, and Martin Bruemmer. Nerd meets nif: Lifting nlp extraction results to the linked data cloud. *LDOW*, 937, 2012.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. *9th LREC*, 2014.

Sara Rodríguez-Fernández, Roberto Carlini, Luis Espinosa-Anke, and Leo Wanner. Example-based Acquisition of Fine-grained Collocation Resources. In *Proceedings of LREC*, Portoroz, Slovenia, 2016.

Stephen Roller, Katrin Erk, and Gemma Boleda. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014*, Dublin, Ireland, 2014.

Sascha Rothe and Hinrich Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of ACL*, pages 1793–1803, Beijing, China, July 2015. Association for Computational Linguistics.

Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

Stefan Rüd, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of ACL-HLT*, pages 965–975, Portland, Oregon, USA, 2011.

Pablo Ruiz and Thierry Poibeau. El92: Entity linking combining open source annotators via weighted voting. In *9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 355–359, 2015.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15, Mexico City, Mexico, 2002.

Bahar Salehi, Paul Cook, and Timothy Baldwin. A word embedding approach to predicting the compositionality of multiword expressions. In *NAACL-HTL*, pages 977–983, Denver, Colorado, 2015.

Gerard Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. ISSN 0001-0782.

Helmut Schmid. Improvements In Part-of-Speech Tagging With an Application To German. In *Proc. of the ACL SIGDAT-Workshop*, pages 47–50, 1995.

Lenhart Schubert. Turing's Dream and the Knowledge Challenge. In *Proc. of AAAI*, pages 1534–1538, 2006.

Hinrich Schütze. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123, 1998.

Hinrich Schütze and Jan Pedersen. Information retrieval based on word senses. In *Proceedings of SDAIR'95*, pages 161–175, Las Vegas, Nevada, 1995.

Roy Schwartz, Roi Reichart, and Ari Rappoport. Symmetric pattern based word embeddings for improved word similarity prediction. *CoNLL 2015*, pages 258âĂŞ–267, 2015.

Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. Learning semantic textual similarity with structural representations. In *Proceedings of ACL (2)*, pages 714–718, Sofia, Bulgaria, 2013.

Hui Shen, Razvan Bunescu, and Rada Mihalcea. Coarse to fine grained sense disambiguation in wikipedia. *Proceedings of *SEM*, pages 22–31, 2013.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of ACL*, pages 2389–2398, 2016.

Ravi Sinha and Rada Mihalcea. Unsupervised graph-based Word Sense Disambiguation using measures of word semantic similarity. In *Proceedings of ICSC*, pages 363–369, 2007.

Frank Smadja. Retrieving Collocations from Text: X-Tract. *Computational Linguistics*, 19(1):143–177, 1993.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, pages 1297–1304, Cambridge, Mass., 2005. MIT Press.

Rion Snow, Dan Jurafsky, and Andrew Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proc. of COLING-ACL 2006*, pages 801–808, 2006.

Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1005–1014, Prague, Czech Republic, 2007.

Benjamin Snyder and Martha Palmer. The English all-words task. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3), Barcelona, Spain*, pages 41–43, Barcelona, Spain, 2004.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. Parsing with compositional vector grammars. In *Proceedings of EMNLP*, pages 455–465, Sofia, Bulgaria, 2013.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. Inverted indexing for cross-lingual NLP. In *Proceedings of ACL*, pages 1713–1722, 2015.

Pascal Soucy and Guy W Mineau. Beyond tfidf weighting for text categorization in the vector space model. In *Proceedings of IJCAI*, volume 5, pages 1130–1135, 2005.

Robert Speer and Joanna Lowry-Duda. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 76–80, 2017.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A core of semantic knowledge. In *Proc. of WWW-07*, pages 697–706, 2007a.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A core of semantic knowledge. unifying WordNet and Wikipedia. In *Proceedings of the 16th World Wide Web Conference,* Banff, Canada, 8–12 May 2007, pages 697–706, 2007b.

Simon Šuster, Ivan Titov, and Gertjan van Noord. Bilingual learning of multi-sense embeddings with discrete autoencoders. In *Proceedings of NAACL-HLT*, pages 1346–1356, 2016.

Kaveh Taghipour and Hwee Tou Ng. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of CoNLL*, pages 338–344, 2015a.

Kaveh Taghipour and Hwee Tou Ng. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of NAACL-HLT 2015*, pages 314–323, 2015b.

L. Tan, H. Zhang, C. Clarke, and M. Smucker. Lexical comparison between wikipedia and twitter corpora by using word embeddings. In *Proceedings of ACL (2)*, pages 657–661, Beijing, China, 2015.

Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Porceedings of EMNLP*, pages 1422–1432, Lisbon, Portugal, 2015.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, 2016.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING*, pages 151–160, 2014.

Jörg Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In *Proc. of LREC*, pages 2214–2218, 2012.

Rocco Tripodi and Marcello Pelillo. A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, 43(1):31–70, 2017.

Yulia Tsvetkov and Shuly Wintner. Identification of multiword expressions by combining multiple linguistic information sources. *Computational Linguistics*, 40 (2):449–468, 2014.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proceedings of EMNLP (2)*, pages 2049–2054, Lisbon, Portugal, 2015.

Dan Tufiş, Radu Ion, Luigi Bozianu, Alexandru Ceauşu, and Dan Ştefănescu. Romanian wordnet: Current state, new applications and prospects. In *Proceedings of 4th Global WordNet Conference, GWC*, pages 441–452, 2008.

Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*, pages 417–424, 2002.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

Amos Tversky and Itamar Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154, 1982.

Shyam Upadhyay, Kai-Wei Chang, James Zou, Matt Taddy, and Adam Kalai. Beyond bilingual: Multi-sense word embeddings using multilingual context. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada, 2017.

Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides GM Petrakis, and Evangelos E Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the 7th annual*

ACM international workshop on Web information and data management, pages 10–16. ACM, 2005.

Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. Evaluating Variants of the Lesk Approach for Disambiguating Words. In *LREC*, 2004.

Paola Velardi, Stefano Faralli, and Roberto Navigli. OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3): 665–707, 2013.

Amir Veyseh and Ben Pouran. Cross-lingual question answering using common semantic space. In *Proceedings of TextGraphs-10: the Workshop on Graph-based Methods for Natural Language Processing*, pages 15–19, 2016.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. Word sense disambiguation for machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 771–778, Vancouver, Canada, 2005.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Thuy Vu and D Stott Parker. K-embeddings: Learning conceptual embeddings for words using context. In *Proceedings of NAACL-HLT*, pages 1262–1267, 2016.

Ivan Vulić and Anna Korhonen. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of ACL*, pages 247–257, 2016.

Yogarshi Vyas and Marine Carpuat. Sparse Bilingual Word Representations for Cross-lingual Lexical Entailment. In *Proceddings of NAACL-HLT*, pages 1187–1197, 2016.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*, pages 1591–1601, 2014.

Leo Wanner, Margarita Alonso Ramos, and Antonia Martí. Enriching the Spanish EuroWordNet by Collocations. In *LREC*, 2004.

Leo Wanner, Bernd Bohnet, and Mark Giereth. Making Sense of Collocations. *Computer Speech and Language*, 20(4):609–624, 2006.

Leo Wanner, Gabriela Ferraro, and Pol Moreno. Towards Distributional Semantics-based Classification of Collocations for Collocation Dictionaries. *International Journal of Lexicography, doi:10.1093/ijl/ecw002*, 2016.

William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. Structured training for neural network transition-based parsing. In *Proceedings of ACL*, pages 323–333, Beijing, China, 2015.

Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. Multi-Objective Optimization for the Joint Disambiguation of Nouns and Named Entities. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 596–605, Beijing, China, 2015a.

Dirk Weissenborn, Feiyu Xu, and Hans Uszkoreit. Dfki: Multi-objective optimization for the joint disambiguation of entities and nouns & deep verb sense disambiguation. *9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 335–339, 2015b.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of EMNLP*, pages 1366–1371, Seattle, Washington, USA, 2013.

Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In *Proceedings of ACL*, pages 118–127, 2010.

Zhaohui Wu and C Lee Giles. Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia. In *Proceedings of AAAI*, pages 2188–2194. Citeseer, 2015.

Yijun Xiao and Kyunghyun Cho. Efficient character-level document classification by combining convolution and recurrent layers. *CoRR*, abs/1602.00367, 2016.

Deyi Xiong and Min Zhang. A sense-based translation model for statistical machine translation. In *Proceedings of ACL*, pages 1459–1469, 2014.

Shweta Yadav, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. Entity extraction in biomedical corpora: An approach to evaluate word embedding features with pso based feature selection. In *Proceedings of EACL*, pages 1159–1170, Valencia, Spain, 2017.

Yadollah Yaghoobzadeh and Hinrich Schütze. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of ACL*, pages 236–246, 2016.

Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, and Gerhard Weikum. Robust question answering over the web of linked data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1107–1116. ACM, 2013.

Hui Yang and Jamie Callan. A metric-based framework for automatic taxonomy induction. In *Proceedings of ACL/IJCNLP*, pages 271–279, 2009.

Yiming Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1-2):69–90, 1999.

Jay Young, Valerio Basile, Lars Kunze, Elena Cabrio, and Nick Hawes. Towards lifelong object learning by integrating situated robot perception and semantic web mining. In *Proceedings of the European Conference on Artificial Intelligence conference*, pages 1458–1466, The Hague, Netherland, 2016.

Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of ACL (2)*, pages 545–550, 2014.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. Learning term embeddings for hypernymy identification. In *Proceedings of IJCAI*, pages 1390–1397, 2015.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING*, pages 1374–1385, 2016.

Zhi Zhong and Hwee Tou Ng. It Makes Sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proceedings of the ACL System Demonstrations*, pages 78–83, 2010.

Zhi Zhong and Hwee Tou Ng. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 273–282. Association for Computational Linguistics, 2012.

Will Y. Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*, pages 1393–1398, 2013.