SAPIENZA
UNIVERSITÀ DI ROMA

Neural-grounded Semantic Representations and Word Sense Disambiguation: A Mutually Beneficial Relationship

Department of Computer Science

Dottorato di Ricerca in Informatica – XXX Ciclo

Candidate

Ignacio Iacobacci
ID number 1645641

Thesis Advisor

Prof. Roberto Navigli

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

September 2018

Thesis defended on 29 February 2019
in front of a Board of Examiners composed by:

Prof. Lamberto Ballan (Università degli Studi di Padova)
Prof. Michele Boreale (Università di Firenze)
Prof. Valeria Cardellini (Università degli Studi di Roma "Tor Vergata")

The thesis has been peer-reviewed by:
Prof. Chris Biemann (Universität Hamburg)
Prof. Danushka Bollegala (University of Liverpool)

Neural-grounded Semantic Representations and Word Sense Disambiguation: A Mutually Beneficial Relationship
Ph.D. thesis. Sapienza – University of Rome

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

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

Para vos, gordo.

Abstract

Language, in both the written and the oral forms, is the ground basis of living in society. The same basic kinds of rules and representations are shared across all the languages. Understand those rules is the objective of Natural Language Processing (NLP), the computerized discipline responsible to analyze and generate language. Building complex computational systems that mimic the human language and are capable to interact and collaborate with us is the holy grail of Natural Language Processing. Semantic representations are the rock-solid foundation on which many successful applications of NLP depend. Their main purpose is to extract and highlight the most important semantic features of textual data. Whereas over the years different approaches have been presented, lately, embeddings have become the dominant paradigm on vectorial representation of items. Currently, many outstanding NLP tasks rely on embeddings to achieve their performance. Embeddings are semantic spaces that carry valuable syntactic and semantic information. The name groups a set of feature learning techniques based on neural networks. Concretely, these techniques are capable to learn semantic spaces that effectively represent words as low-dimensional continuous vectors. They also maintain the structure of language by representing diverse lexical and semantic relations by a relation-specific vector offset. With the increasing amount of available text, as well as the increased computing power, techniques which take advantage of large volumes of unstructured data, as word embeddings, have become the prevailing approach of semantic representation of natural language. However, despite their enormous success, common word-embeddings approaches came with two inherent flaws: these representations are incapable to handle ambiguity, as senses of polysemous words are aggregated into single vectors. In addition, most word embeddings rely only on statistical information of word occurrences, leaving aside existing rich knowledge of structured data. To tackle the problem of polysemy, a fundamental task of Natural Language Processing (NLP), Word Sense Disambiguation (WSD), seems particularly suitable. The task, an open problem in the discipline, aims at identifying the correct meaning of word based given its context. Concretely, it links each word occurrence to a sense from a predefined inventory. Most successful approaches for WSD combine the use of unstructured data, manually annotated datasets and semantic resources.

In the present thesis we address the issue of of ambiguity in semantic representations from a multimodal perspective. Firstly, we introduce and investigate new neural-based approaches to build better word and sense embeddings relying on both statistical data and prior semantic knowledge. We employ diverse techniques of WSD for linking word occurrences to their correct meaning on large amounts of raw corpora. Then, we use the resulting data as training input for learning the embeddings. We show the quality of these representations by evaluating them on standard semantic similarity frameworks reporting state-of-the-art performance on multiple datasets. Secondly, we show how these representations are capable to create better WSD systems. We introduce a new way to leverage word representations which outperforms current WSD approaches in both supervised and unsupervised configurations. We show that our WSD framework, based solely on embeddings, is capable to surpass WSD approaches based on standard features. Thirdly, we propose two new technique for leveraging semantic-annotated data. We incorporate more semantic features resulting in an increment in the performance compared with our initial approaches. We close the loop by showing that our semantic representations enhanced with WSD are also suitable for improving the task of WSD itself.

Acknowledgments

To my advisor Roberto Navigli, for giving me this tremendous opportunity from the other side of the World, trusting in me and pushing me hard in order to get the best out of me. To my colleagues from the Linguistic Computing Laboratory: those who started with me, José Camacho-Collados, Claudio Delli Bovi and Alessandro Raganato, those who share with me only a portion of the PhD time, Daniele Vanella, Tiziano Flati, Andrea Moro and especially to Mohammad Taher Pilehvar who was a colleage, a friend and an advisor. Those who came after me, Tommaso Pasini, Valentina Pyatkin, Federico Scozzafava, Andrea Di Fabio, Martina Piromalli and Marco Maru. I have learned from each one of you. To my mentors in Argentina, who encourage me to start working on Natural Language Processing, José Castaño, Ernesto Mislej and Fernando Balbachan. To my Master's director, Enrique C. Segura with whom I entered the wonderful world of neural networks. To Michael Zock, who has been a great counselor. To my family, my brother Mariano, my sister Daniela and especially my mother Marta, who have supported me from afar each in their own way. Finally to my lovely wife Leticia, without her advice and company, nothing of this would have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Our Language is very very important. That's us and makes us"

Jerry Wolfe

"... language circumscribes beauty, confines it, limns, delineates, colours and contains. Yet what is language but a tool, a tool we use to dig up the beauty that we take as our only and absolute real"

"A Bit of Fry & Laurie" show

The main purpose of language is to share our thoughts and feelings. It allows us to work collaboratively, to ask for help, to live in society. Language is part of our identity. Nations are based, and named, by the language spoken by their citizens. We, as individuals living in community, utilize spoken or written forms of language to organize each other and share ideas. Language affects the way we take decisions. We tend to choose options more emotionally when a question is formulated in our mother tongue while using a foreign language reduces decision-making biases (Keysar et al., 2012). Some controversial theories go even further and consider that language determines the structure of how the real world is perceived by human beings, and that this structure is different from one language to another (Hoijer, 1954). Early ancestors of humanity used some kind of gestural language to hunt in groups and organize the herd. Evidence suggests that around 200 thousand years ago, Homo sapiens learned how to speak and to communicate via spoken language.

The irruption of writing, dated approximately around 5,000 years ago, is the most important technology in human life (Powell, 2012). Written language allowed humans to avoid depending upon memory of the messenger, since a statement could be left written to be read by another person which was not present at the time of writing it. Also, Language has been used, instead, to restrict the potential receiver of a message. Examples are Navajo, a native American language used to communicate encrypted messages between allied troops (Kawano, 1990); or Nüshu, an endangered language based on Chinese script known only by women (Fasold and Connor-Linton, 2014).

Nowadays, there are around 7000 different languages, many of them spoken by only a few people. Strong efforts are being made to understand them, to make them available to other mother tongue speakers and to discover ancient ones. The scientific discipline responsible to the study of language is known as linguistics. Started as a proper discipline during early 20th century, mostly thanks to the contributions of Noah Chomsky (Chomsky, 1957, 1964), has been evolved during the years. The inclusion of statistical and computerized analysis has created a new discipline which we name today as Natural Language Processing (NLP), which was defined by Liddy (2001) as a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing.

1.1 Ambiguity

Ideas are abstract representations that live in human minds, therefore they cannot be expressed directly. Hence there is a mapping between meanings or thoughts and linguistic forms that we call *language*. Likewise, language is not a static thing. Through time new concepts grow, while others are deprecated and disappear. Words change their use, relations between them evolve, changing between meanings and becoming more dominant senses barely used in the past. This constant evolution makes it difficult to have a direct mapping between ideas and words or sentences. In addition, in spoken language, there is a limit in the amount of different sounds or phonemes that we can produce and understand. Therefore, we have opted to utilize a set of compositional rules to make the language grow. These rules allow us to combine low-level units, such as phonemes or morphemes, into higher order units, such as words or sentences. This idea is known in the literature with the name of duality of patterning (Hockett and Hockett, 1960), and refers to the fact that the meaningful units of language are made up of meaningless units whose only function is to distinguish the meaningful units from one another.

Nonetheless, the mapping between ideas and language is not unequivocal. Single concepts can be expressed with different words, which we call synonyms (like *buy* and *purchase*), while individual words may represent a set of different meanings. If the meanings are unrelated we identify such words as homonyms. An example of this is the word *suit*, which refers both to a clothing and to a proceeding in a court of law. On the other hand, if the word's meanings are related or have a common origin, we call that word polyseme, such as *chicken*, which refers to both the animal and the food, or *camera*, which refers both to legislative chamber, and device for recording visual images, derived from *camera obscura*, the precursor of photography.

This imperfect mapping between words and concepts constitutes the problem of lexical ambiguity. But this ambiguity is not a flaw in the language and evidence suggests that well-designed communication systems are inherently ambiguous. Ambiguity exists for reasons of communicative efficiency (Piantadosi et al., 2012). At human level, ambiguity is not a problem, since we use common knowledge to understand each other. In fact, some studies have found little evidence for active use of ambiguity-avoidance strategies (Arnold et al., 2004; Roland et al., 2006).

A system responsible to analyze and generate language should be able to determine the meaning of an ambiguous word in particular context, just as we humans do.

1.2 Semantic Spaces

A fundamental piece of language understanding is the study concerned with the meaning of a word. The branch of linguistics that deals with words and the concepts that they represent was named, in ancient times, semasiology, from Greek sēmasia ("meaning") and from sēmaínō, ("signify"). Today, the discipline is known as semantics.

"A word is characterized by the company it keeps" it is what Firth claimed in 1957, referring to the idea that the meaning of a given word is based on the context where the word appears. The Distributional Hypothesis, claimed by Harris (1954), says that words that are used and occur in the same contexts tend to purport similar meanings. Distributional Semantics is the branch of semantics which has a fundamental base in the distributional hypothesis which represents words as vectors derived from the statistical information of their co-occurrence context in a large corpus. The set of vectors is commonly named as *Semantic Space*. A Semantic Space is a space, often with a large number of dimensions, in which words or concepts are represented by vectors; the position of each such point along each axis is somehow related to the meaning of the word (Morris, 1958).

Semantic spaces are useful to examine the relationships between the words or concepts within them because, once the space is built, relationships can be quantified by applying distance metrics to those vectors. Semantic spaces were traditionally constructed by first defining the meanings of a set of axes and then gathering information from human subjects to determine where each word in question should fall on each axis. The continuous evolution of languages fits perfectly in a Semantic Space which represents words and their meaning as vectors, and their relations as transformations in the space. A proper representation of meaning is fundamental for a complete understanding of language.

Many long-standing approaches are able to learn semantic spaces from text corpora. Latent Semantic Analysis (Deerwester et al., 1990) takes advantage of implicit higher-order structure in the association of terms with documents ("semantic structure") in order to improve the detection of relevant documents on the basis of terms found in queries. This approach, originally created for Information Retrieval (Latent Semantic Indexing), creates a large term-to-document matrix, which, by using singular-value decomposition, shrinks the matrix to around 100 features. Word vectors are created based on word occurrences through documents. A newer approach, Hyperspace Analogue to Language (HAL), was introduced by Lund and Burgess (1996). The basic premise of HAL is that words with similar meaning repeatedly occur closely (namely co-occur). As an example in a large corpus of text one could expect to see the words mouse, dog and cat appear often close to each other. The same might be true for Japan, Malaysia and Singapore. More recently, Latent Dirichlet Allocation (Blei et al., 2003, LDA) presented a generative probabilistic model for collections of discrete data such as text corpora. LDA, created originally for Topic Modelling, is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities.

The introduction of word2vec[1] (Mikolov et al., 2013b) brought to a complete disruption in NLP. The newly introduced tool was able to learn effective representations of words from large amounts of text corpora. The representations learned from this kind of neural-based models received the named of Word Embedding. Their great quality was noticed immediately. The representations appear to project satisfactorily the inherent structure of language, as lexical and semantic relations between words are represented with a relation-specific vector offset. Word2vec introduced two different models, namely Continuous Bag-of-Words (CBOW), and SkipGram, both based on a simple feed-forward neural language model. CBOW is a two-layer network where the projection layer averages all the context words and a log-linear classifier, on top of that, infers the word in the middle given its neighboring words. The second architecture, Skip-Gram, instead, predicts the surrounding words given the central one. Both architectures showed outstanding results in many NLP tasks.

1.3 Word Sense Disambiguation

Usual Semantic Spaces come with an inherent flaw: word-based representations are typically built by conflating all the senses of each word into a single vector. This compound representation is extremely harmful to representing them, since related words of different senses tend be similarly represented, hence related in terms of the semantic space. This is evident in the case of the word *bar*. The common sense tend to associate bar mostly with a commercial establishment where alcoholic drinks are served. But *bar* is also used for naming an elongated piece of metal or wood. So a word-based representation will represent with similar vectors *beer*, an alcoholic drink deeply related with the first sense of bar, and also *rod*, a thin bar.

Word Sense Disambiguation (WSD), defined by Manning and Schütze (1999) in the

[1] `http://code.google.com/p/word2vec/`

following way: "[t]he task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word. This is done by looking at the context of the word's use", is a fundamental task in computational lexical semantics (Navigli, 2009).

One of the open issues of WSD is to choose the right way to represent different meanings. Most efforts represent word meanings with a link to sense inventory. Those approaches use, as sense inventory, either a semantic network as WordNet or BabelNet (Patwardhan and Pedersen, 2006; Guo and Diab, 2010; Ponzetto and Navigli, 2010; Miller et al., 2012; Agirre et al., 2014; Chen et al., 2014) or a Thesaurus (Mohammad and Hirst, 2006; Agirre et al., 2010). In addition, there are alternative ways to represent meanings. Substitution-based approaches, proposed by McCarthy and Navigli (2007), explored the use of synonym or near-synonym lexical substitutions to characterize the meaning of word occurrences. In contrast to dictionary senses, substitutes are not taken to partition a word's meaning into distinct senses. Distributional approaches, instead, represent diverse meanings by dissembling the word senses via clustering techniques that exploit co-occurrence with context features over a large corpus (Reisinger and Mooney, 2010; Huang et al., 2012; Neelakantan et al., 2014; Tian et al., 2014).

Since the final performance of NLP systems depends on the quality of their input representations, a semantic space which represents not only individual words but also its constituent meanings seems indispensable reaching the ultimate goal of NLP: a complete understanding of natural language.

1.4 Semantic Networks

While WSD systems which utilize manually sense annotated data, the so-called supervised approaches, are among the most accurate models, they generally fail with respect to coverage. It's very expensive and time-consuming to create an accurate sense-annotated corpus that sacrifice least common words. To be able to handle large amount of different words, a large sense inventory is fundamental and that is where Semantic Networks are the dominant in WSD tasks.

As was defined by Lehmann (1992), "A semantic network is a graph of the structure of meaning". In general, a semantic network is a graph which represents knowledge in patterns of interconnected nodes and arcs. Several efforts in NLP were done in order to create larger and better semantic networks. The most widely used Semantic Networks is WordNet (Miller, 1995). WordNet a manually curated lexical database of English, inspired by psycholinguistic theories of human lexical memory. In WordNet, English nouns, verbs, adjectives and adverbs are organized into *synsets*, sets of synonyms effectively representing unique concepts. Each synset to which a word belong is a possible sense of that word. For instance, the noun *plane* in WordNet 3.1 is linked to the synsets "plane, sheet" and "aeroplane, plane, airplane" (among others), defining the two distinct senses of *plane* as an abstract concept and a mean of transportation respectively. Synsets in WordNet have unique identifiers, a plain text definition, and they are organized in a taxonomy according to the relation of hypernymy.

Despite its size (version 3.1 counts over 117.000 synsets) and widespread usage, WordNet is hindered by a few limitations. In particular, it only covers the English language and it does not support named entities such as people and locations. Freebase Bollacker et al. (2008) was a large online collaborative knowledge base consisting of structured data structured data from many sources. The objective of Freebase was to create a global resource capable to offer more effective access to common information. It was developed by the American software company Metaweb and ran publicly since March 2007. Freebase is now deprecated and currently is part of Google's Knowledge Graph.

Another well known resource is BabelNet (Navigli and Ponzetto, 2012), a multilingual encyclopedic dictionary and semantic network, including approximately 14 million entries for concepts and named entities linked by semantic relations. Like in WordNet, BabelNet concepts are organized in synsets, each containing the words that express their concept in a wide range of languages. BabelNet was originally created by merging WordNet and Wikipedia[2], thus augmenting the manually curated lexical knowledge of the former with the large scale, collaboratively built general knowledge of the latter, and grew over time to include several other resources such as Wiktionary[3] and OmegaWiki[4].

Semantic Networks are particularly useful to perform non-supervised WSD. Examples

[2] `https://www.wikipedia.org/`

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

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

of this were presented in Agirre and Soroa (2009), who propose a new graph-based method which use the knowledge in WordNet to perform WSD by utilizing PageRank Brin and Page (1998) on the whole graph. Another approach is Moro et al. (2014) who perform Personalized PageRank (a variant of PageRank centered in a particular node) over the whole graph of BabelNet.

1.5 Objectives

The main objectives of this thesis are:

- To create a complete semantic space of words and concepts, which leverage information from semantic networks and surpass the current state of semantic representations.
- To leverage semantic spaces in order to create better WSD systems which outperform state-of-the-art performance in standard benchmarks.
- To take advantage of diverse neural network architectures to create better semantic representations aided with prior semantic knowledge.
- To make available the both the representations and the tools which we used to create them and make exploitable by the research community.

1.6 Contributions and Published material

A wide portion of the ideas presented in this thesis has already been published in top NLP conferences. In the following we list these publications. The content of some of these publications represent the core of this thesis and are included at great extent in some chapters and sections, and indicated accordingly below:

- Chapter 2 introduces the related work about semantic representation of items, putting special emphasis in the polysemy issue and how was tackled by diverse approaches.
- In Chapter 3 and Chapter 4 we introduced two new models that leverage WSD models for building rich semantic representations.

In Chapter 3 we introduce SensEmbed a neural approach which leverages semantic knowledge to obtain continuous representations for individual word senses for effective semantic similarity measurement. This chapter covers the publication of Iacobacci et al. (2015) published as oral presentation in the Proceedings of the 53nd Annual Meeting of the Association for the Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015). The research described in this chapter was carried out in its entirely by myself. The other authors in the publication acted as advisers.

In Chapter 4, we propose a model that jointly learns word and sense embeddings and represents them in a unified vector space by exploiting large corpora and knowledge obtained from semantic networks. The relevant publication of this chapter is Mancini et al. (2017), published the proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017). I acted as co-advisor in this research paper and my contributions were mostly on the architecture of the introduced model. The shallow disambiguation strategy (Section 4.2) and implementation details were developed with the other authors.

- In Chapter 5, we go through the reverse path, by putting forward a model for WSD, taking advantage of the semantic information carried by embeddings. The relevant publication of this chapter is Iacobacci et al. (2016), published the proceedings of the 54st Annual Meeting of the Association for the Computational Linguistics (ACL 2016).

- In the following Chapters 6 and 7 we extend further the approach of SensEmbed introduced in Chapter 3.

  Chapter 6 introduces SensEmbed+, a joint distributional and knowledge-based approach for obtaining low-dimensional continuous representations for word, word senses and synsets which leverages the learned representations and lexical-semantic knowledge. We put forward a framework for many NLP tasks such as word, sense, and relational similarity and Word Sense Disambiguation with state-of-the-art performance on multiple datasets. Part of this chapter has been submitted to Computational Linguistics as a journal paper and is still under review.

Chapter 7 presents new way to learn embeddings utilizing a Recurrent Neural Network (RNN) based on a bidirectional Long Short Term Memory (LSTM) for learning continuous representations of word sense, taking into account word ordering. In addition we present a new idea to enrich these representations with semantic knowledge from large corpora and vocabularies. Part of this chapter has been submitted to the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) and is still under review.

- Chapter 8 includes conclusions and closing remarks of the presented research.

The rest of the thesis contains novel, unpublished material which has been added to improve the clarity of the presentation and strengthen the relationships between the parts.

Chapter 2

Preliminaries: Semantic Representation of items

"The unavoidable implication is that a sense of "similarity" must be innate [..] This "stimulus generalization" happens automatically, without extra training, and it entails an innate "similarity space" [..] These subjective spacings of stimuli are necessary for learning.."

Steven Pinker, 1994

Machine-interpretable representations of lexical and semantic items are key for a number of NLP tasks, and therefore obtaining good representations is an important goal of the research in NLP, as shown by the surge of recent work on this topic. A summary of approaches to the representation of lexical and semantic items is given in this chapter.

2.1 Embeddings

In mathematical terms, given two structures A and B, A is said to be *embedded* in B if there exists an injective function $\alpha : A \to B$ which is *structure preserving*, according to some property of the specific structures involved. The idea of embedding has been applied to NLP by defining the concept of *word embedding*, an embedding of the space of words (that is, the vocabulary of natural language) into a continuous vector space. As such, a word embedding model is a function that associates each word in a given language with a vector. Most importantly, by being structure preserving, word embeddings encode syntactic and semantic information, such as interesting regularities of the natural language, and represent relationships between words, with geometric relations between their corresponding vectors.

2.2 Word Embeddings

Many long-standing approaches are able to learn real-valued vector representations from text. Early models such as Latent Semantic Analysis (Deerwester et al., 1990, LSA), Hyperspace Analogue to Language (Lund and Burgess, 1996, HAL) and Latent Dirichlet Allocation (Blei et al., 2003, LDA), mentioned in Section 1.2, exploit information of word co-occurrence, representing words as vectors of co-occurrences with documents, words and topics respectively, to generate a Semantic Space where words with similar meanings are represented in closer proximity.

Advances of Neural-based model for word representation began to attract the attention of the NLP community. Actual word embeddings were first introduced by Bengio et al. (2003). His initial model was based on a multilayer perceptron (MLP) with two hidden layers: a shared non-linear and a regular hidden hyperbolic tangent one. The goal of this network was learning the joint probability function of a sequence of words, i.e. statistical language modeling, i.e. Further developments as the *hierarchical log-bilinear* model (Mnih and Hinton, 2007, HLBL), a probabilistic linear neural model which aims to predict the embedding of a word given the concatenation of the previous words; or Collobert and Weston (2008), which deepened the original neural model by adding a convolutional layer and an extra layer for modeling long-distance dependencies increased even more the efforts in that direction.

The contribution made by Mikolov et al. (2013b) meant a groundbreaking event in the discipline. The introduced model, called *word2vec*, simplified the original architecture from Bengio et al. (2003) by removing the hyperbolic tangent layer and hence significantly speeding up the training process. In addition, they showed that word representations learned with a neural network trained on raw text geometrically encode highly latent relationships. The canonical example is the vector resulting from $king - man + woman$ found to be very close to the induced vector of $queen$. GloVe (Pennington et al., 2014), an alternative approach trained on aggregated global word-word co-occurrences, obtained similar results. This new efforts allowed the use of vast amounts of raw text as training data. These approaches allowed the community to create denser and more effective representations.

On top of those, newer efforts were presented in order to improve the representations. Some incorporated prior semantic knowledge to enrich embeddings both during learning, such as Yu and Dredze (2014) who presented an alternative way to train word embeddings, based on word2vec, by using, in addition to common features, words having some relation in a semantic resource, like PPDB[1] or WordNet[2]. Faruqui et al. (2015)[3], instead, presented a technique applicable to pre-processed embeddings, in which vectors are updated ("retrofitted") in order to make them more similar to those which share a word type and less similar to those which do not. The word types were extracted from diverse semantic resources such as PPDB, WordNet and FrameNet[4]. Other approaches took a different path. For instance the model from Mitchell and Steedman (2015), who decompose the word embeddings based on the syntactic and semantic properties of their corresponding words, or include linguistic information in the learning process such as the word's part-of-speech and its position like Qiu et al. (2014) and Liu et al. (2016). Press and Wolf (2017) introduced another model based on word2vec, where the embeddings are extracted from the output topmost weight matrix, instead of the input one, showing that those representations constitute also a valid word embedding. Finally, two different which are worth mentioning Vilnis and McCallum (2014), who modeled words as density in the vector space, rather than single point, and the work of Nalisnick and Ravi (2017) who modeled instead embeddings with arbitrary dimensionality.

While these embeddings are surprisingly good for monosemous words, they fail to represent the non-dominant senses of words properly due to the pervasive sense skewness. This makes that neighboring vectors of representations of ambiguous words are overwhelmed by its dominant sense. As for the word *plane*, its close vicinity only has words related to the

[1] http://www.paraphrase.org/#/download

[2] https://wordnet.princeton.edu/

[3] https://github.com/mfaruqui/retrofitting

[4] https://framenet.icsi.berkeley.edu/

aircraft meaning, such as *airplane*, *jet* or *flight*, letting aside words related to the geometrical meaning (unbounded two-dimensional shape) or spiritual meaning (a level of existence or development). Since most commonly-used words have several meanings, a model which attends to ambiguity is needed for an effective semantic representation.

2.3 Going to the Sense level

Recent advances in word representations, like those that we mentioned above, allowed the community to create denser and more effective representations, where relations between words are represented by an offset in the vector space. Nevertheless, word embeddings do not explicitly distinguish between different meanings of a word since it conflates all of its word senses in a single vector.

To address the polysemy issue, sense embeddings represent individual word senses as separate vectors. Learning sense embeddings is a prolific research area with many efforts in that direction. Two main approaches have been put forward in the literature:

- the **unsupervised approach**, in which meanings are acquired automatically as a result of discriminating occurrences within text. The resulting vectors are usually referred to as multi-prototype representations;

- the **knowledge-based approach**, which relies on existing sense inventories like WordNet, Freebase, Wikipedia or BabelNet, to name a few.

Below we introduce models corresponding to both both approaches.

2.3.1 Multi-prototype Embeddings

The unsupervised approach allows to create a complete unsupervised system. The discrimination of senses relies on statistics based on co-occurrences from text corpora. Due to the ability to be trained without any human intervention, this approach is ideal for being implemented on neural-network-based end-to-end models. Numerous unsupervised efforts have been presented so far: Reisinger and Mooney (2010), for instance, created a vector space model of word meanings without using any sense inventory. Pointing out that a single prototype is incapable of capturing homonymy and polysemy, they presented a method that

Model	Corpus	Bilingual	# Senses
Reisinger and Mooney (2010)	Wikipedia & Gigaword		Specified
Huang et al. (2012)	Wikipedia		Specified
Neelakantan et al. (2014)	Wikipedia		Determined by method
Tian et al. (2014)	Wikipedia		Specified
Guo et al. (2014)	LDC03E24, LDC04E12 & IWSLT 2008, PKU 863	✓	Specified
Li and Jurafsky (2015)	Wikipedia & Gigaword		Determined by method
Wu and Giles (2015)	Wikipedia		Specified
Melamud et al. (2016)	UKWAC		Contextualized Word Embeddings
Lee and Chen (2017)	Wikipedia		Determined by method
Athiwaratkun and Wilson (2017)	UKWAC & Wackypedia		Specified
McCann et al. (2017)	Multi30k, WMT 2017 IWSLT 2016 MT Task	✓	Contextualized Word Embeddings
Peters et al. (2018)	1 Billion Word Benchmark[5]		Contextualized Word Embeddings

Table 2.1. Multi-prototype approaches

leverages clustering techniques to produce multiple "sense-specific" (they called like that) vector for each word. In their learning process, word's contexts are clustered to produce groups of similar contexts vectors. An average prototype vector is computed for each cluster producing a set of vectors for each word. The dimensions of the vectors represent a co-occurrence between the word and other words present in the training corpus; Huang et al. (2012) adopted a similar strategy by decomposing each word's single-prototype representation into multiple prototypes, denoting different senses of that word. To this end, they first gathered the context for all occurrences of a word and then used spherical K-means to cluster the contexts. Each cluster was taken as the context for a specific meaning of the word and hence used to train embeddings for that specific meaning (i.e., word sense); Neelakantan et al. (2014) presented an extension of the Skip-gram model of word2vec capable to learn multiple embeddings for a single word. Moreover, the model has no assumption about the number of prototypes, and all the representations are learned in a single step speeding up the learning process. This approach comes in two flavors. The first one, Multi-Sense Skip-gram (MSSG) which have a specific pre-defined number of prototypes per word, and Non-parametric Multi-Sense Skip-gram (NP-MSSG) which, uses a threshold to induce a new sense vector. Tian et al. (2014) exploited pre-trained embeddings and considered them as learned from word occurrences generated by a mixture of a finite number of different word senses. Their main idea was to combine the robustness of the Skip-gram model, with just a few parameters to adjust and based on local context of raw text and the framework that a mixture model provide, avoiding the extra step that a clustering needs. Guo et al.

(2014) exploited parallel data to automatically generate sense-annotated data, based on the fact that different senses of a word are usually translated to different words in another language (Chan and Ng, 2005). The automatically-generated sense-annotated data was later used for training sense-specific word embeddings; Wu and Giles (2015) introduced a model called Sense-aware semantic analysis (SaSa), which takes into account the idea of using the global context in order to distinguish meanings of the occurrences of the words by producing "sense-specific" prototypes by clustering Wikipedia pages. Similarly to the work of Reisinger and Mooney (2010), their vector features are explicit but instead of using word, they used concepts, representing different Wikipedia articles. Li and Jurafsky (2015) used a Chinese restaurant process as a way to induce new senses. For each new word occurrence during the training the current word could either "sit" at one of the existing tables (corresponding to one of the previously seen senses) or choose a new table (a new sense). The decision is made by measuring semantic relatedness (based on local context information and global document information) and the number of customers already sitting at that table (the popularity of word senses). Lee and Chen (2017), an unsupervised approach that introduced a modularized framework to create sense-level representation learned with linear-time sense selection. Athiwaratkun and Wilson (2017), who, extending the word-based approach of Vilnis and McCallum (2014), decomposed word vectors as a mixture of densities of their constituent senses.

Alternative approaches exploited recurrent neural networks (RNN) architectures to their learning process and created representations for both words and contexts. Melamud et al. (2016) introduced context2vec, a model based on a bidirectional Long Short Term Memory (LSTM), a particular type of RNN for learning sentence and word embeddings. They use large raw text corpora to learn a neural model that embeds entire sentential contexts and target words in the same low-dimensional space. McCann et al. (2017) presented CoVe, an approach for transferring knowledge from an LSTM-based encoder pretrained on machine translation. They used the outputs of the LSTM as context vectors and applied to a variety of downstream NLP tasks. Peters et al. (2018) presented ELMo, a word-in-context representation model based in a deep bidirectional language model. Just as CoVe, ELMo has not a token dictionary, but each token is represented by three vector, two of which being contextual. In Table 2.1 we highlight the salient features of each multi-prototype mentioned above.

Nevertheless, representations are not devoid of issues. Since the induced vectors are discriminated based on statistics, low-frequency senses or uncertain contexts are generally underrepresented, if they are represented at all. The resulting inventory tends to be coarser than knowledge-based approaches and biased towards more frequent senses. In addition, the fact that the representations are not linked to any sense inventory makes their evaluation more difficult.

2.3.2 Sense Embeddings

Knowledge-based approaches, on the other hand, need some general-purpose human intervention, typically for the creation of the meaning inventory and its structure, such as a thesaurus, an ontology or a semantic network. One of the earliest works on knowledge-based sense representations was the so-called Salient Semantic Analysis (Hassan and Mihalcea, 2011, SSA). SSA used Wikipedia pages as its sense inventory and exploited the hyperlinks through other pages as a sense-annotated corpus. The representations where calculated as an explicit vector of co-occurring Wikipedia pages.

As examples of efforts which rely on WordNet we can name the work of Chen et al. (2014)[6], who leveraged word embeddings in Word Sense Disambiguation and investigated the possibility of retrofitting embeddings with the resulting disambiguated words. Chen et al. (2015), presented a novel approach by the word sense embeddings through learning sentence-level embeddings from WordNet glosses. Their approach is an extension of the Neelakantan et al. (2014) model which initializes the multiprototype vectors of the MGGS with those leared from the WordNet Glosses. Jauhar et al. (2015) introduced a multi-sense approach based on expectation-maximization style algorithms for inferring word sense choices in the training corpus and learning sense embeddings while incorporating ontological sources of information such as WordNet. In AutoExtend[7] (Rothe and Schütze, 2015), instead, sense vectors are inferred in the same semantic space of pre-trained word embeddings. Their approach learns embeddings for lexemes, senses and synsets from WordNet in a shared space by inferring embeddings given the constraint that word representations are sums of their lexeme representations and synset representations are sums of the representations of

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

[7] http://cistern.cis.lmu.de/~sascha/AutoExtend/

Model	Inventory	Annotated	Corpus
Hassan and Mihalcea (2011)	Wikipedia	✓	Wikipedia
Bordes et al. (2013)	FreeBase	-	
Chen et al. (2014)	WordNet	Wikipedia	
Wang et al. (2014b)	Freebase	-	
Wang et al. (2014a)	Freebase	-	
Chen et al. (2015)	WordNet	✓	WordNet Glosses
Jauhar et al. (2015)	WordNet	WMT-2011 English	
Rothe and Schütze (2015)	WordNet	Google News	
Flekova and Gurevych (2016)	WordNet Supersenses	✓	Wikipedia
Lin et al. (2015)	Freebase	-	
Pilehvar and Collier (2016)	WordNet	Google News	
Camacho-Collados et al. (2016)	BabelNet	✓	Wikipedia
Nickel et al. (2016)	Freebase	-	

Table 2.2. Sense Embeddings approaches.

the lexemes they contain. Flekova and Gurevych (2016) presented a distributional joint model of words and supersenses. Supersenses are coarse-grained semantic categories defined in WordNet in which all the items of the inventory fall. The defined categories contain 26 labels (such as ANIMAL, PERSON or FEELING) for nouns and 15 labels for verbs (such as COMMUNICATION, MOTION or COGNITION). DeConf[8] (Pilehvar and Collier, 2016), also linked to WordNet, presented a model where sense vectors are inferred in the same semantic space of pre-trained word embeddings by decomposing the given word representation into its constituent senses. Further, Camacho-Collados et al. (2016) introduced NASARI[9], an approach which similarly, exploited Wikipedia for the creation semantic vector representations but instead used as its sense inventory the corresponding BabelNet synsets and Wikipedia pages in several languages. Finally, several approaches aimed to create a semantic space without relying on distributional information, such as Bordes et al. (2013), Wang et al. (2014b), Wang et al. (2014a), Lin et al. (2015) and Nickel et al. (2016) among others. These approaches created a semantic space by leveraging the graph structure of Freebase. Table 2.2 shows the nature of each sense embeddings approach mentioned, the training corpus and the sense inventory used.

None of these approaches fully exploit both distributional and structured information (e.g., by covering both named entities and common nouns together with verbs, adjectives and adverbs while taking advantage of large amounts of raw text). Since both WordNet and

[8] https://pilehvar.github.io/deconf/

[9] http://lcl.uniroma1.it/nasari/

Wikipedia are contained in BabelNet, an approach linked to that resource, and where lexical and semantic items are represented together, is desirable. In the following chapters we will introduce several approaches of semantic representation of items. We show that by taking advance of Word Sense Disambiguation, while leveraging a powerful knowledge resource as BabelNet, and large amounts of text corpora, we can build better and more informative embeddings, which, in turn, can be leveraged for the task of WSD. We perform several experiments reaching state-of-the-art performance in several standard benchmarks.

Chapter 3

SENSEMBED: Learning Sense Embeddings for Word and Relational Similarity

> "...if [..] one can see not only the central word in question, but also say N words on either side, then, if N is large enough one can unambiguously decide the meaning of the central word"
>
> *Warren Weaver, 1949*

3.1 Summary

Word embeddings have recently gained considerable popularity for modeling words in different Natural Language Processing (NLP) tasks including semantic similarity measurement. However, notwithstanding their success, word embeddings are by their very nature unable to capture polysemy, as different meanings of a word are conflated into a single representation. In addition, their learning process usually relies on massive corpora only, preventing them from taking advantage of structured knowledge. We address both issues by proposing a multi-faceted approach that transforms word embeddings to the sense level and leverages

knowledge from a large semantic network for effective semantic similarity measurement. We evaluate our approach on word similarity and relational similarity frameworks, reporting state-of-the-art performance on multiple datasets.

3.2 Introduction

The much celebrated word embeddings represent a new branch of corpus-based distributional semantic model which leverages neural networks to model the context in which a word is expected to appear. Thanks to their high coverage and their ability to capture both syntactic and semantic information, word embeddings have been successfully applied to a variety of NLP tasks, such as Word Sense Disambiguation (Chen et al., 2014), Machine Translation (Mikolov et al., 2013b), Relational Similarity (Mikolov et al., 2013c), Semantic Relatedness (Baroni et al., 2014) and Knowledge Representation (Bordes et al., 2013).

However, as we mentioned in Chapter 2.2, word embeddings inherit two important limitations from their antecedent corpus-based distributional models: (1) they are unable to model distinct meanings of a word as they conflate the contextual evidence of different meanings of a word into a single vector; and (2) they base their representations solely on the distributional statistics obtained from corpora, ignoring the wealth of information provided by existing semantic resources.

Several research works have tried to address these problems. For instance, basing their work on the original sense discrimination approach of Reisinger and Mooney (2010), Huang et al. (2012) applied K-means clustering to decompose word embeddings into multiple prototypes, each denoting a distinct meaning of the target word. However, the sense representations obtained are not linked to any sense inventory, a mapping that consequently has to be carried out either manually, or with the help of sense-annotated data. Another line of research investigates the possibility of taking advantage of existing semantic resources in word embeddings. A good example is the Relation Constrained Model (Yu and Dredze, 2014). When computing word embeddings, this model replaces the original co-occurrence clues from text corpora with the relationship information derived from The Paragraph Database (Ganitkevitch et al., 2013, PPDB).

However, none of these techniques have simultaneously solved both above-mentioned

bank$^n_1$	***bank***$^n_2$	***number***$^n_4$	***number***$^n_3$	***hood***$^n_1$	***hood***$^n_{12}$
(geographical)	(financial)	(phone)	(acting)	(gang)	(convertible car)
upstream$^r_1$	commercial_bank$^n_1$	calls$^n_1$	appearing$^v_6$	tortures$^n_5$	taillights$^n_1$
downstream$^r_1$	financial_institution$^n_1$	dialed$^v_1$	minor_roles$^n_1$	vengeance$^n_1$	grille$^n_2$
runs$^v_6$	national_bank$^n_1$	operator$^n_{20}$	stage_production$^n_1$	badguy$^n_1$	bumper$^n_2$
confluence$^n_1$	trust_company$^n_1$	telephone_network$^n_1$	supporting_roles$^n_1$	brutal$^a_1$	fascia$^n_2$
river$^n_1$	savings_bank$^n_1$	telephony$^n_1$	leading_roles$^n_1$	execution$^n_1$	rear_window$^n_1$
stream$^n_1$	banking$^n_1$	subscriber$^n_2$	stage_shows$^n_1$	murders$^n_1$	headlights$^n_1$

Table 3.1. Closest senses to two senses of three ambiguous nouns: *bank*, *number*, and *hood*

issues, i.e., inability to model polysemy and reliance on text corpora as the only source of knowledge. We propose a novel approach, called SENSEMBED, which addresses both drawbacks by exploiting semantic knowledge for modeling arbitrary word senses in a large sense inventory. We evaluate our representation on multiple datasets in two standard tasks: word-level semantic similarity and relational similarity. Experimental results show that moving from words to senses, while making use of lexical-semantic knowledge bases, makes embeddings significantly more powerful, resulting in consistent performance improvement across tasks. Our contributions are twofold: (1) we propose a knowledge-based approach for obtaining continuous representations for individual word senses; and (2) by leveraging these representations and lexical-semantic knowledge, we put forward a semantic similarity measure with state-of-the-art performance on multiple datasets.

3.3 Sense Embeddings

Word embeddings are a vector space models (VSM) that represent words as real-valued vectors in a low-dimensional (relative to the size of the vocabulary) semantic space, usually referred to as the continuous space language model. In contrast to word embeddings, which obtain a single vector for potentially ambiguous words, sense embeddings are continuous representations of individual word senses.

In order to be able to apply word embeddings techniques to obtain representations for individual word senses, large sense-annotated corpora have to be available. However, manual sense annotation is a difficult and time-consuming process, i.e., the so-called knowledge acquisition bottleneck. In fact, the largest existing manually sense annotated dataset is the

SemCor corpus (Miller et al., 1993), whose creation dates back to more than two decades ago. In order to alleviate this issue, we leveraged a state-of-the-art WSD algorithm to automatically generate large amounts of sense-annotated corpora.

The chapter continues as follows: in Section3.3.1, we describe the sense inventory used for SENSEMBED. Section 3.3.2 introduces the corpus and the disambiguation procedure used to sense annotate this corpus. Finally in Section 3.3.3 we discuss how we leverage the automatically sense-tagged dataset for the training of sense embeddings.

3.3.1 Underlying sense inventory

We selected BabelNet (cf. Section 1.4) as our underlying sense inventory. The resource is a merger of WordNet with multiple other lexical resources, the most prominent of which is Wikipedia. As a result, the manually-curated information in WordNet is augmented with the complementary knowledge from collaboratively-constructed resources, providing a high coverage of domain-specific terms and named entities and a rich set of relations. The usage of BabelNet as our underlying sense inventory provides us with the advantage of having our sense embeddings readily applicable to multiple sense inventories.

3.3.2 Generating a sense-annotated corpus

As our corpus we used the September-2014 dump of the English Wikipedia.[1] This corpus comprises texts from various domains and topics and provides a suitable word coverage. The unprocessed text of the corpus includes approximately three billion tokens and more than three million unique words. We only consider tokens with at least five occurrences.

As our WSD system, we opted for Babelfy[2] (Moro et al., 2014), a state-of-the-art WSD and Entity Linking algorithm based on BabelNet's semantic network. Babelfy first models each concept in the network through its corresponding “semantic signature" by leveraging a graph random walk algorithm. Given an input text, the algorithm uses the generated semantic signatures to construct a subgraph of the semantic network representing the input text. Babelfy then searches this subgraph for the intended sense of each content word using an iterative process and a dense subgraph heuristic. Thanks to its use of BabelNet,

[1] http://dumps.wikimedia.org/enwiki/

[2] http://www.babelfy.org/

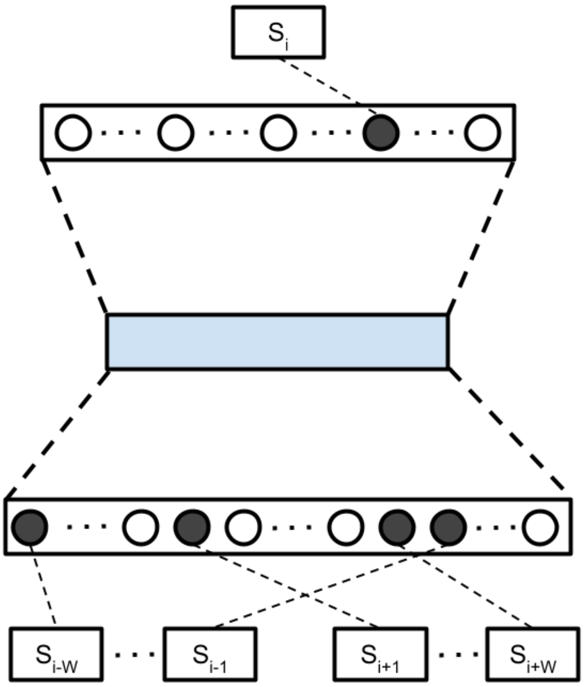

Figure 3.1. The SensEmbed architecture.

Babelfy inherently features multilinguality; hence, our representation approach is equally applicable to languages other than English. In order to guarantee high accuracy and to avoid bias towards more frequent senses, we do not consider those judgements made by Babelfy while backing off to the most frequent sense, a case that happens when a certain confidence threshold is not met by the algorithm. The disambiguated items with high confidence correspond to more than 50% of all the content words. As a result of the disambiguation step, we obtain sense-annotated data comprising around one billion tagged words with at least five occurrences and 2.5 million unique word senses.

3.3.3 Learning sense embeddings

The disambiguated text is processed with the word2vec (Mikolov et al., 2013a) toolkit. We applied word2vec to produce continuous representations of word senses based on the distributional information obtained from the annotated corpus. For each target word sense, a representation is computed by maximizing the log likelihood of the word sense with respect to its context. We opted for the Continuous Bag of Words (CBOW) architecture, the objective of which is to predict a single word (word sense in our case) given its context.

	Synset Description	**Synonymous senses**
hood_1^n	rough or violent youth	hoodlum_1^n, goon_2^n, thug_1^n
hood_4^n	photography equipment	lens_hood_1^n
hood_9^n	automotive body parts	bonnet_2^n, cowl_1^n, cowling_1^n
hood_{12}^n	car with retractable top	convertible_1^n

Table 3.2. Sample initial senses of the noun *hood* (leftmost column) and their synonym expansion.

The context is defined by a window, typically with the size of five words on each side with the paragraph ending barrier. We used hierarchical softmax as our training algorithm. The dimensionality of the vectors were set to 400 and the sub-sampling of frequent words to 10^{-3}.

As a result of the learning process, we obtain vector-based semantic representations for each of the word senses in the automatically-annotated corpus. We show in Table 3.1 some of the closest senses to six sample word senses: the geographical and financial senses of *river*, the performance and phone number senses of *number*, and the gang and car senses of *hood*.[3] As can be seen, sense embeddings can capture effectively the clear distinctions between different senses of a word. Additionally, the closest senses are not necessarily constrained to the same part of speech. For instance, the river sense of *bank* has the adverbs *upstream* and *downstream* and the "move along, of liquid" sense of the verb *run* among its closest senses.

3.4 Similarity Measurement

This Section describes how we leverage the generated sense embeddings for the computation of word similarity and relational similarity. We start the Section by explaining how we associate a word with its set of corresponding senses and how we compare pairs of senses in Sections 3.4.1 and 3.4.2, respectively. We then illustrate our approach for measuring word similarity, together with its knowledge-based enhancement, in Section 3.4.3, and relational

[3] We follow Navigli (2009) and show the n^{th} sense of the *word* with part of speech x as $\textit{word}_n^x$.

similarity in Section 3.4.4. Hereafter, we refer to our similarity measurement approach as SENSEMBED.

3.4.1 Associating senses with words

In order to be able to utilize our sense embeddings for a word-level task such as word similarity measurement, we need to associate each word with its set of relevant senses, each modeled by its corresponding vector. Let $\mathcal{S}_w$ be the set of senses associated with the word w. Our objective is to cover as many senses as can be associated with the word w. To this end we first initialize the set $\mathcal{S}_w$ by the word senses of the word w and all its synonymous word senses, as defined in the BabelNet sense inventory. We show in Table 3.2 some of the senses of the noun *hood* and the synonym expansion for these senses. We further expand the set $\mathcal{S}_w$ by repeating the same process for the lemma of word w (if not already in lemma form).

3.4.2 Vector comparison

To compare vectors, we use the *Tanimoto* distance. The measure is a generalization of Jaccard similarity for real-valued vectors in [-1, 1]:

$$\mathcal{T}(\vec{w_1}, \vec{w_2}) = \frac{\vec{w_1} \cdot \vec{w_2}}{\|\vec{w_1}\|^2 + \|\vec{w_2}\|^2 - \vec{w_1} \cdot \vec{w_2}} \tag{3.1}$$

where $\vec{w_1} \cdot \vec{w_2}$ is the dot product of the vectors $\vec{w_1}$ and $\vec{w_2}$, and $\|\vec{w_i}\|$ is the Euclidean norm of $\vec{w_i}$. Rink and Harabagiu (2013) reported consistent improvements when using vector space metrics, in particular the Tanimoto distance, on the SemEval-2012 task on relational similarity (Jurgens et al., 2012) in comparison to several other measures that are designed for probability distributions, such as Jensen-Shannon divergence and Hellinger distance.

3.4.3 Word similarity

We show in Algorithm 1 our procedure for measuring the semantic similarity of a pair of input words w_1 and w_2. The algorithm also takes as its inputs the similarity strategy and the *weighted* similarity parameter α (Section 3.4.3) along with a *graph vicinity factor* flag (Section 3.4.3).

Similarity measurement strategy

We take two strategies for calculating the similarity of the given words w_1 and w_2. Let $\mathcal{S}_{w_1}$ and $\mathcal{S}_{w_2}$ be the sets of senses associated with the two respective input words w_1 and w_2, and let $\vec{s_i}$ be the sense embedding vector of the sense s_i. In the first strategy, which we refer to as *closest*, we follow the conventional approach of Budanitsky and Hirst (2006) and measure the similarity of the two words as the similarity of their closest senses, i.e.:

$$Sim_{\mathit{closest}}\left(w_1, w_2\right) = \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} \mathcal{T}\left(\vec{s_1}, \vec{s_2}\right) \tag{3.2}$$

However, taking the similarity of the closest senses of two words as their overall similarity ignores the fact that the other senses can also contribute to the process of similarity judgement. In fact, psychological studies suggest that humans, while judging semantic similarity of a pair of words, consider different meanings of the two words and not only the closest ones (Tversky, 1977; Markman and Gentner, 1993). For instance, the WordSim-353 dataset (Finkelstein et al., 2002) contains the word pair *brother-monk*. Despite having the religious devotee sense in common, the two words are assigned the similarity judgement of 6.27, which is slightly above the middle point in the similarity scale [0,10] of the dataset. This clearly indicates that other non-synonymous, yet still related, senses of the two words have also played a role in the similarity judgement. Additionally, the relatively low score reflects the fact that the religious devotee sense is not a dominant meaning of the word *brother*.

We therefore put forward another similarity measurement strategy, called *weighted*, in which different senses of the two words contribute to their similarity computation, but the contributions are scaled according to their relative importance. To this end, we first leverage sense occurrence frequencies in order to estimate the dominance of each specific word sense. For each word w, we first compute the dominance of its sense $s \in \mathcal{S}_w$ by dividing the frequency of s by the overall frequency of all senses associated with w, i.e., $\mathcal{S}_w$:

$$d(s) = \frac{freq(s)}{\sum_{s' \in \mathcal{S}_w} freq(s')} \tag{3.3}$$

We further recognize that the importance of a specific sense of a word can also be

Algorithm 1 Word Similarity

Input: Two words w_1 and w_2
Str, the similarity strategy
Vic, the *graph vicinity factor* flag
α parameter for the *weighted* strategy
Output: The similarity between w_1 and w_2

```
$\mathcal{S}_{w_1} \leftarrow$ getSenses$(w_1)$
$\mathcal{S}_{w_2} \leftarrow$ getSenses$(w_2)$
if Str is closest then
    sim $\leftarrow$ -1
else
    sim $\leftarrow 0$
end if
for each $s_1 \in \mathcal{S}_{w_1}$ and $s_2 \in \mathcal{S}_{w_2}$ do
    if Vic is true then
        tmp $\leftarrow \mathcal{T}^*(\vec{s_1},\vec{s_2})$
    else
        tmp $\leftarrow \mathcal{T}(\vec{s_1},\vec{s_2})$
    end if
    if Str is closest then
        sim $\leftarrow \max$ (sim, tmp)
    else
        sim $\leftarrow$ sim + tmp$^{\alpha} \times d(s_1) \times d(s_2)$
    end if
end for
```

triggered by the word it is being compared with. We model this by biasing the similarity computation towards closer senses, by increasing the contribution of closer senses through a power function with parameter α. The similarity of a pair of words w_1 and w_2 according to the *weighted* strategy is computed as:

$$Sim_{\textit{weighted}}\,(w_1, w_2) = \sum_{s_1 \in \mathcal{S}_{w_1}} \sum_{s_2 \in \mathcal{S}_{w_2}} d(s_1)\; d(s_2)\; \mathcal{T}(\vec{s_1}, \vec{s_2})^{\alpha} \tag{3.4}$$

where the α parameter is a real-valued constant greater than one. We show in Section 3.5.1 how we tune the value of this parameter.

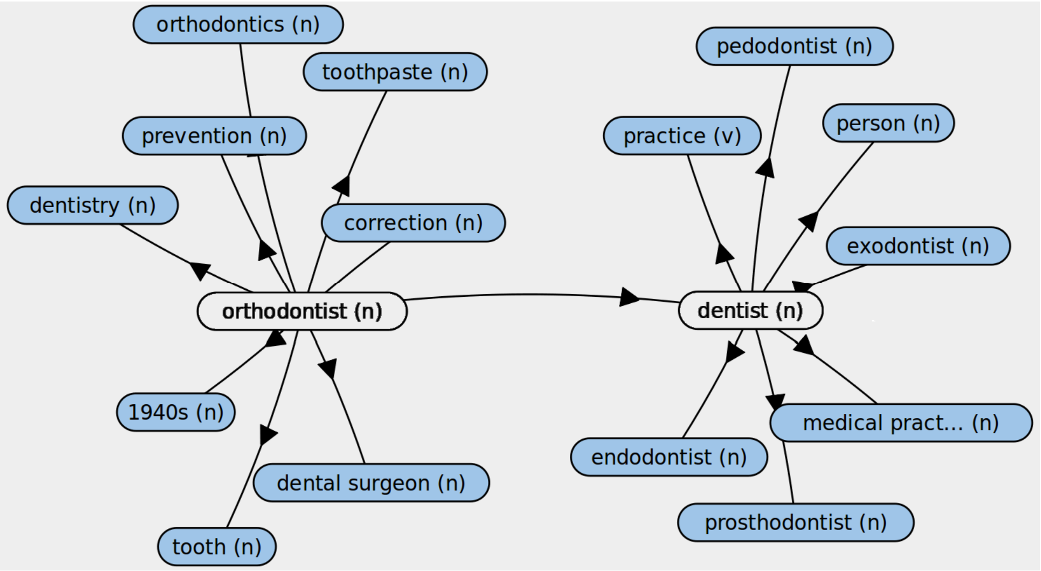

Figure 3.2. Portion of BabelNet graph including the pair *orthodontist-dentist* and close related synsets

Enhancing similarity accuracy

Our similarity measurement approach takes advantage of lexical knowledge at two different levels. First, as we described in Sections 3.3.2 and 3.3.3, we use a knowledge-based disambiguation approach, i.e., Babelfy, which exploits BabelNet's semantic network. Second, we put forward a methodology that leverages the relations in BabelNet's graph for enhancing the accuracy of similarity judgements, to be discussed next.

As a distributional vector representation technique, our sense embeddings can potentially suffer from inaccurate modeling of less frequent word senses. In contrast, our underlying sense inventory provides a full coverage of all its concepts, with relations that are taken from WordNet and Wikipedia. In order to make use of the complementary information provided by our lexical knowledge base and to obtain more accurate similarity judgements, we introduce a *graph vicinity factor*, that combines the structural knowledge from BabelNet's semantic network and the distributional representation of sense embeddings. To this end, for a given sense pair, we scale the similarity judgement obtained by comparing their corresponding sense embeddings, based on their placement in the network. Let E be the set of all sense-to-sense relations provided by BabelNet's semantic network, i.e., $E = \{(s_i, s_j) : s_i - s_j\}$. Then, the similarity of a pair of words with the *graph vicinity factor* in formulas 3.2 and 3.4

is computed by replacing $\mathcal{T}$ with $\mathcal{T}^*$, defined as:

$$\mathcal{T}^*(\vec{s_1}, \vec{s_2}) = \begin{cases} \mathcal{T}(\vec{s_1}, \vec{s_2}) \times \beta, & \text{if } (s_1, s_2) \in E \\ \mathcal{T}(\vec{s_1}, \vec{s_2}) \times \beta^{-1}, & \text{otherwise} \end{cases} \tag{3.5}$$

We show in Section 3.5.1 how we tune the parameter β. This procedure is particularly helpful for the case of less frequent word senses that do not have enough contextual information to allow an effective representation. For instance, the SimLex-999 dataset (Hill et al., 2015), which we use as our tuning dataset (see Section 3.5.1), contains the highly-related pair *orthodontist-dentist*, which is also connected in the BabelNet graph (Figure 3.2). We observed that the intended sense of the noun *orthodontist* occurs only 70 times in our annotated corpus. As a result, the obtained representation was not accurate, resulting in a low similarity score for the pair. The two respective senses are, however, directly connected in the BabelNet graph. Hence, the *graph vicinity factor* scales up the computed similarity value for the word pair.

3.4.4 Relational similarity

Relational similarity evaluates the correspondence between relations (Medin et al., 1990). The task can be viewed as an analogy problem in which, given two pairs of words (w_a, w_b) and (w_c, w_d), the goal is to compute the extent to which the relations of w_a to w_b and w_c to w_d are similar. Sense embeddings are suitable candidates for measuring this type of similarity, as they represent relations between senses as linear transformations. Given this property, the relation between a pair of words can be obtained by subtracting their corresponding normalized embeddings. Following Zhila et al. (2013), the relational similarity between two pairs of word (w_a, w_b) and (w_c, w_d) is accordingly calculated as:

$$\text{ANALOGY}(\vec{w_a}, \vec{w_b}, \vec{w_c}, \vec{w_d}) = \mathcal{T}(\vec{w_b} - \vec{w_a}, \vec{w_d} - \vec{w_c}) \tag{3.6}$$

We show the procedure for measuring the relational similarity in Algorithm 2. The algorithm first finds the closest senses across the two word pairs: s_a^* and s_b^* for the first pair and s_c^* and s_d^* for the second. The analogy vector representations are accordingly computed as the difference between the sense embeddings of the corresponding closest senses. Finally, the

Algorithm 2 Relational Similarity

```
Input: Two pairs of words $w_a$, $w_b$ and $w_c$, $w_d$
Output: The degree of analogy between the two pairs

$\mathcal{S}_{w_a} \leftarrow getSenses(w_a)$, $\mathcal{S}_{w_b} \leftarrow getSenses(w_b)$
$(s_a^*, s_b^*) \leftarrow argmax_{\substack{s_a \in \mathcal{S}_{w_a} \\ s_b \in \mathcal{S}_{w_b}}} \mathcal{T}(\vec{s_a}, \vec{s_b})$
$\mathcal{S}_{w_c} \leftarrow getSenses(w_c)$, $\mathcal{S}_{w_d} \leftarrow getSenses(w_d)$
$(s_c^*, s_d^*) \leftarrow argmax_{\substack{s_c \in \mathcal{S}_{w_c} \\ s_d \in \mathcal{S}_{w_d}}} \mathcal{T}(\vec{s_c}, \vec{s_d})$
return: $\mathcal{T}(\vec{s_b}^* - \vec{s_a}^*, \vec{s_d}^* - \vec{s_c}^*)$
```

relational similarity is computed as the similarity of the analogy vectors of the two pairs.

3.5 Experiments

We evaluate our sense-enhanced semantic representation on multiple word similarity and relatedness datasets (Section 3.5.1), as well as the relational similarity framework (Section 3.5.2).

3.5.1 Word similarity experiment

Word similarity measurement is one of the most popular evaluation methods in lexical semantics, and semantic similarity in particular, with numerous evaluation benchmarks and datasets. Given a set of word pairs, a system's task is to provide similarity judgments for each pair, and these judgments should ideally be as close as possible to those given by humans.

Datasets

We evaluate SENSEMBED on standard word similarity and relatedness datasets: the RG-65 (Rubenstein and Goodenough, 1965) and the WordSim-353 (Finkelstein et al., 2002, WS-353) datasets. Agirre et al. (2009) suggested that the original WS-353 dataset conflates similarity and relatedness and divided the dataset into two subsets, each containing pairs for just one type of association measure: similarity (the WS-Sim dataset) and relatedness (the WS-Rel dataset).

Measure	**Dataset**					
	RG-65	WS-Sim	WS-Rel	YP-130	MEN	Avg
Pilehvar et al. (2013)	0.868	0.677	0.457	0.710	0.690	0.677
Zesch et al. (2008)	0.820	—	—	0.710	—	—
Collobert and Weston (2008)	0.480	0.610	0.380	—	0.570	—
word2vec (Baroni et al., 2014)	0.840	0.800	0.700	—	0.800	—
GloVe	0.769	0.666	0.559	0.577	0.763	0.737
ESA	0.749	—	—	—	—	—
PMI-SVD	0.738	0.659	0.523	0.337	0.726	0.695
word2vec	0.732	0.707	0.476	0.343	0.665	0.644
SensEmbed$_{closest}$	**0.894**	0.756	0.645	**0.734**	0.779	0.769
SensEmbed$_{weighted}$	0.871	**0.812**	**0.703**	0.639	**0.805**	**0.794**

Table 3.3. Spearman correlation performance on five word similarity and relatedness datasets.

We also evaluate our approach on the YP-130 dataset, which was created by Yang and Powers (2005) specifically for measuring verb similarity, and also on the Stanford's Contextual Word Similarities (SCWS), a dataset for measuring word-in-context similarity (Huang et al., 2012). In the SCWS dataset each word is provided with the sentence containing it, which helps in pointing out the intended sense of the corresponding target word.

Finally, we also report results on the MEN dataset which was recently introduced by Bruni et al. (2014). MEN contains two sets of English word pairs, together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk.

Comparison systems

We compare the performance of our similarity measure against twelve other approaches. As regards traditional distributional models, we report the best results computed by Baroni et al. (2014) for PMI-SVD, a system based on Pointwise Mutual Information (PMI) and SVD-based dimensionality reduction. For word embeddings, we report the results of GloVe (Pennington et al., 2014) and Collobert and Weston (2008). GloVe is an alternative way for learning embeddings, in which vector dimensions are made explicit, as opposed to the opaque meaning of the vector dimensions in word2vec. The approach of Collobert and Weston (2008) is an embeddings model with a deeper architecture, designed to preserve more complex knowledge as distant relations. We also show results for the word embeddings trained by Baroni et al. (2014). The authors first constructed a massive corpus by combining several large corpora. Then, they trained dozens of different word2vec models by varying the system's training parameters and reported the best performance obtained on each dataset.

As representatives for graph-based similarity techniques, we report results for the state-of-the-art approach of Pilehvar et al. (2013) which is based on random walks on WordNet's semantic network. Moreover, we present results for the graph-based approach of Zesch et al. (2008), which compares a pair of words based on the path lengths on Wiktionary's semantic network. We also compare our word similarity measure against the multi-prototype models of Reisinger and Mooney (2010) and Huang et al. (2012), and against the approaches of Yu and Dredze (2014) and Chen et al. (2014), which enhance word embeddings with semantic knowledge derived from The Paragraph Database (PPDB) and WordNet, respectively. Finally, we report results for word embeddings, as our baseline, obtained using the word2vec toolkit on the same corpus that was annotated and used for learning our sense embeddings (cf. Section 3.3.3).

Parameter tuning

Recall from Sections 3.4.3 and 3.4.3 that our algorithm has two parameters: the α parameter for the *weighted* strategy and the β parameter for the *graph vicinity factor*. We tuned these two parameters on the SimLex-999 dataset (Hill et al., 2015). We picked SimLex-999 since there are not many comparison systems in the literature that report results on the dataset. We found the optimal values for α and β to be 8 and 1.6, respectively.

Results

Table 3.3 shows the experimental results on five different word similarity and relatedness datasets. We report the Spearman correlation performance for the two strategies of our approach as well as eight other comparison systems. SENSEMBED proves to be highly reliable on both similarity and relatedness measurement tasks, obtaining the best performance on most datasets. In addition, our approach shows itself to be equally suitable for verb similarity, as indicated by the results on YP-130.

The rightmost column in the Table shows the average performance weighted by dataset size. Between the two similarity measurement strategies, *weighted* proves to be the more

Measure	WS-353	SCWS
Huang et al. (2012)	0.713	0.628
Reisinger and Mooney (2010)	0.770	–
Chen et al. (2014)	–	**0.662**
Yu and Dredze (2014)	0.537	–
word2vec	0.694	0.642
SENSEMBED$_{closest}$	0.714	0.589
SENSEMBED$_{weighted}$	**0.779**	0.624

Table 3.4. Spearman correlation performance of the multi-prototype and semantically-enhanced approaches on the WordSim-353 and the Stanford's Contextual Word Similarities datasets.

suitable, achieving the best overall performance on three datasets and the best mean performance of 0.794 across the two strategies. This indicates that our assumption of considering all senses of a word in similarity computation was beneficial.

We report in Table 3.4 the Spearman correlation performance of four approaches that are similar to SENSEMBED: the multi-prototype models of Reisinger and Mooney (2010) and Huang et al. (2012), and the semantically enhanced models of Yu and Dredze (2014) and Chen et al. (2014). We provide results only on WS-353 and SCWS, since the above-mentioned approaches do not report their performance on other datasets. As we can see from the Table, SENSEMBED outperforms the other approaches on the WS-353 dataset. However, our approach lags behind on SCWS, highlighting the negative impact of taking the closest senses as the intended meanings. In fact, on this dataset, SENSEMBED$_{weighted}$ provides better performance owing to its taking into account other senses as well. The better performance of the multi-prototype systems can be attributed to their coarse-grained sense inventories which are automatically constructed by means of Word Sense Induction.

3.5.2 Relational similarity experiment

Dataset and evaluation. We take as our benchmark the SemEval-2012 task on Measuring Degrees of Relational Similarity (Jurgens et al., 2012). The task provides a dataset comprising 79 graded word relations, 10 of which are used for training and the rest for test. The task evaluated the participating systems in terms of the Spearman correlation and the MaxDiff score (Louviere, 1991).

Measure	MaxDiff	Spearman
Com	45.2	0.353
PairDirection	45.2	—
RNN-1600	41.8	0.275
UTD-LDA	—	0.334
UTD-NB	39.4	0.229
UTD-SVM	34.7	0.116
PMI baseline	33.9	0.112
word2vec	43.2	0.288
SENSEMBED$_{closest}$	**45.9**	**0.358**

Table 3.5. Spearman correlation performance of different systems on the SemEval-2012 Task on Relational Similarity.

Comparison systems. We compare our results against six other systems and the PMI baseline provided by the task organizers. As for systems that use word embeddings for measuring relational similarity, we report results for RNN-1600 (Mikolov et al., 2013c) and PairDirection (Levy and Goldberg, 2014). We also report results for UTD-NB and UTD-SVM (Rink and Harabagiu, 2012), which rely on lexical pattern classification based on Naïve Bayes and Support Vector Machine classifiers, respectively. UTD-LDA (Rink and Harabagiu, 2013) is another system presented by the same authors that casts the task as a selectional preferences one. Finally, we show the performance of Com (Zhila et al., 2013), a system that combines word2vec, lexical patterns, and knowledge base information. Similarly to the word similarity experiments, we also report a baseline based on word embeddings (word2vec) trained on the same corpus and with the same settings as SENSEMBED.

Results. Table 3.5 shows the performance of different systems in the task of relational similarity in terms of the Spearman correlation and MaxDiff score. A comparison of the results for word2vec and SENSEMBED shows the advantage gained by moving from the word to the sense level. Among the comparison systems, Com attains the closest performance. However, we note that the system is a combination of several methods, whereas SENSEMBED is based on a single approach.

Model	**Setting**			**Dataset**					
	Strategy	Vicinity	Expansion	RG-65	WS-Sim	WS-Rel	YP-130	MEN	Average
word2vec	–	–		0.732	0.707	0.476	0.343	0.665	0.644
word2vec$_{exp}$	–	–	✓	0.700	0.665	0.326	0.621	0.655	0.632
SensEmbed	*closest*			0.825	0.693	0.488	0.492	0.712	0.690
			✓	0.844	0.714	0.562	0.681	0.743	0.728
		✓	✓	**0.894**	0.756	0.645	**0.734**	0.779	0.769
	weighted			0.877	0.776	0.639	0.446	0.783	0.762
			✓	0.864	0.783	0.665	0.591	0.773	0.761
		✓	✓	0.871	**0.812**	**0.703**	0.639	**0.805**	**0.794**

Table 3.6. Spearman correlation performance of word embeddings (word2vec) and SensEmbed on different semantic similarity and relatedness datasets.

3.5.3 Analysis

In order to analyze the impact of the different components of our similarity measure, we carried out a series of experiments on our word similarity datasets. We show in Table 3.6 the experimental results in terms of Spearman correlation. Performance is reported for the two similarity measurement strategies, i.e., *closest* and *weighted*, and for different system settings with and without the expansion procedure (cf. Section 3.4.1) and *graph vicinity factor* (cf. Section 3.4.3). As our comparison baseline, we also report results for word embeddings, obtained using the word2vec toolkit on the same corpus and with the same configuration (cf. Section 3.3.3) used for learning the sense embeddings (word2vec in the Table). The rightmost column in the Table reports the mean performance weighted by dataset size. word2vec$_{exp}$ is the word embeddings system in which the similarity of the two words is determined in terms of the closest word embeddings among all the corresponding synonyms obtained with the expansion procedure (cf. Section 3.4.1).

A comparison of word and sense embeddings in the vanilla setting (with neither the expansion procedure nor *graph vicinity factor*) indicates the consistent advantage gained by moving from word to sense level, irrespective of the dataset and the similarity measurement strategy. The consistent improvement shows that the semantic information provided more than compensates for the inherently imperfect disambiguation. Moreover, the results indicate the consistent benefit gained by introducing the *graph vicinity factor*, highlighting the fact that our combination of the complementary knowledge from sense embeddings and information derived from a semantic network is beneficial. Finally, note that the expansion procedure leads to performance improvement in most cases for sense embeddings. In direct contrast, the step proves harmful in the case of word embeddings, mainly due to their inability to distinguish individual word senses.

3.6 Conclusions

We propose an approach for obtaining continuous representations of individual word senses, referred to as sense embeddings. Based on the proposed sense embeddings and the knowledge obtained from a large-scale lexical resource, i.e., BabelNet, we put forward an effective technique, called SENSEMBED, for measuring semantic similarity. We evaluated our approach on multiple datasets in the tasks of word and relational similarity. Two conclusions can be drawn on the basis of the experimental results: (1) moving from word to sense embeddings can significantly improve the effectiveness and accuracy of the representations; and (2) a meaningful combination of sense embeddings and knowledge from a semantic network can further enhance the similarity judgements.

Chapter 4

Embedding Words and Senses Together via Joint Knowledge-Enhanced Training

"There is no more striking general fact about language than its universality "

Edward Sapir, 1921

Word embeddings are widely used in Natural Language Processing (NLP), mainly due to their success in capturing semantic information from massive corpora. However, their creation process does not allow the different meanings of a word to be automatically separated, as it conflates them into a single vector. We address this issue by proposing a new model which learns word and sense embeddings jointly. Our model exploits large corpora and knowledge from semantic networks in order to produce a unified vector space of word and sense embeddings. We evaluate the main features of our approach both qualitatively and quantitatively in a variety of tasks, highlighting the advantages of the proposed method in comparison to state-of-the-art word- and sense-based models.

4.1 Introduction

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 (NLP) 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.

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; Mrksic 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 Nieto Piña, 2015; Jauhar et al., 2015; Rothe and Schütze, 2015; Pilehvar and Collier, 2016; Camacho-Collados et al., 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. 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.

4.2 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 3 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 20). 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 6 to 11). 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 12 to 22).[2] This parameter aims to retain relevant connectivity across senses, as only

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

[2] As mentioned above, all unigrams, bigrams and trigrams present in the semantic network are considered. In

Algorithm 3 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 $w \in T$ do
    $S_T \leftarrow S_T \cup S_w$ ($S_w$: set of candidate synsets of $w$)
end for
Minimum connections threshold $\theta \leftarrow \frac{|S_T|+|T|}{2\,\delta}$
Output set of connections $T^* \leftarrow \emptyset$
for each $w \in T$ do
    Relative maximum connections $max = 0$
    Set of senses associated with $w$, $C_w \leftarrow \emptyset$
    for each candidate synset $s \in S_w$ do
        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
                $C_w \leftarrow C_w \cup \{(w, s)\}$
            end if
        end if
    end for
    $T^* \leftarrow T^* \cup C_w$
end for
return Output set of connected words and senses $T^*$
```

senses above the threshold will be connected to words in the output corpus. θ is proportional to the reciprocal of a parameter δ,[3] 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 (3) 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.

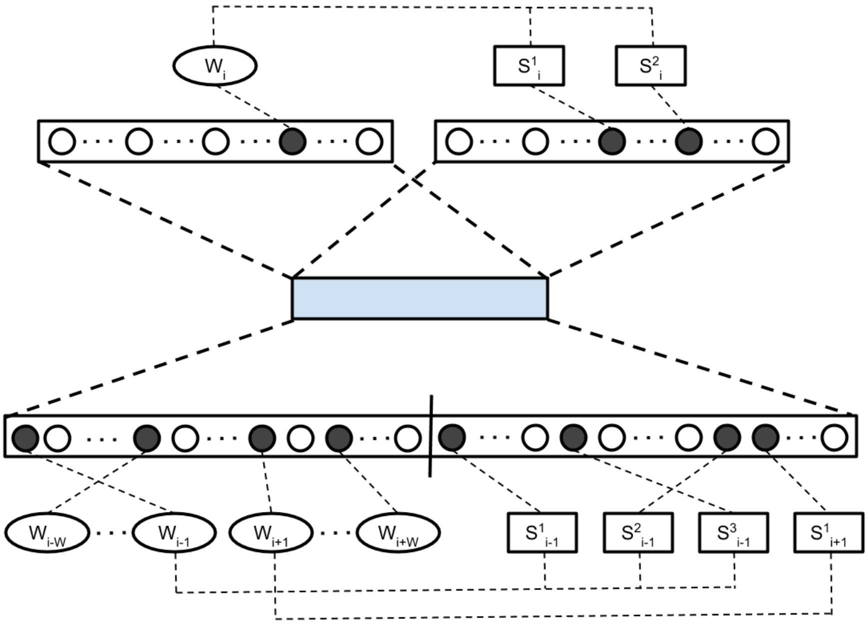

Figure 4.1. The SW2V architecture

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. 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 4.4.2, 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 (Miller, 1995) or BabelNet (Navigli and Ponzetto, 2012) are hampered by the high granularity of their sense inventories (Hovy et al., 2013). In Section 4.5.2 we show how our sense embeddings are particularly suited to deal with this issue.

4.3 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 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 $W = w_{t-n}, ..., w_t, ..., w_{t+n}$ (being w_t the target word) and $S = S_{t-n}, ..., S_t, ..., S_{t+n}$, where $S_i = s_i^1, ..., s_i^{k_i}$ 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 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 4.1 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.

[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.

4.3.1 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.

4.3.2 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.

4.4 Analysis of Model Components

In this section we analyze the different components of SW2V, including the nine model configurations (Section 4.4.1) and the algorithm which generates the connections between

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

words and senses in context (Section 4.4.2). 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 (cf. Section 1.4), a large multilingual semantic network with over 350 million semantic connections, integrating resources such as Wikipedia and WordNet. We chose BabelNet owing to its wide coverage of named entities and lexicographic knowledge.

- **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., 2015). For the analysis we use two word similarity datasets: the similarity portion (Agirre et al., 2009, 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, we apply the standard closest senses strategy (Resnik, 1995; Budanitsky and Hirst, 2006; Camacho-Collados et al., 2015a), using cosine similarity (cos) as comparison measure between senses:

$$sim(w_1, w_2) = \max_{s \in S_{w_1}, s' \in S_{w_2}} \cos(\vec{s}_1, \vec{s}_2) \tag{4.1}$$

[5] `http://ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/`

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.

4.4.1 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 4.3 that our model could have words, senses or both in either the input and output layers. Table 4.1 shows the results of all nine configurations on the WS-Sim and RG-65 datasets.

As shown in Table 4.1, 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 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.

4.4.2 Disambiguation / Shallow word-sense connectivity algorithm

In this section we evaluate the impact of our *shallow word-sense connectivity algorithm* (Section 4.2) 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 Frequent Sense (MFS) heuristic as a back-off strategy and, following the work done in SensEmbed (cf. Chapter 3.3), we also include a version in which only instances above the Babelfy default confidence threshold are disambiguated (i.e. the MFS back-off strategy is disabled). We will refer to this latter

[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.

[7] http://babelfy.org

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

Table 4.2. Pearson (r) and Spearman (ρ) correlation performance of SW2V integrating our *shallow* word-sense connectivity algorithm (default),Babelfy and Babelfy+MFS.

version as Babelfy+MFS and report the best configuration of each strategy according to our analysis.

Table 4.2 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.

4.5 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 in the word similarity task (Section 4.5.1). Second, we measure the quality of our sense embeddings in a sense-specific application: sense clustering (Section 4.5.2). Finally, we evaluate the coherence of our unified vector space by measuring the interconnectivity of word and sense embeddings (Section 4.5.3).

Experimental setting. Throughout all the experiments we use the same standard hyperparameters mentioned in Section 4.4 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

Type	**Model**	**Corpus**	**Dataset**			
			SimLex-999		**MEN**	
			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 4.3. Pearson (r) and Spearman (ρ) correlation performance on the SimLex-999 and MEN word similarity datasets.

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. We compare with the publicly available pre-trained sense embeddings of four state-of-the-art models: Chen et al. (2014) and AutoExtend (Rothe and Schütze, 2015) based on WordNet, and SensEmbed (cf. Chapter 3.3) and NASARI (Camacho-Collados et al., 2016) based on BabelNet.

4.5.1 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[8]. SimLex 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 4.4, 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) using WordNet (Retrofitting$_{\mathrm{WN}}$) and BabelNet (Retrofitting$_{\mathrm{BN}}$) as lexical resources.

Table 4.3 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*).[9] In contrast to the consistently low gold similarity scores given to antonym pairs, our system varies its similarity scores depending on the specific nature of the pair[10]. 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; Mrksic et al., 2017), but this

[8] To enable a fair comparison we did not perform experiments on the small datasets used in Section 4.4 for validation.

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

[10] 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.

is outside the scope of this chapter.

4.5.2 Sense Clustering

Current lexical resources tend to suffer from the high granularity of their sense inventories (Palmer et al., 2007). In fact, a meaningful clustering of their senses may lead to improvements on downstream tasks (Hovy et al., 2013; Flekova and Gurevych, 2016; Pilehvar et al., 2017). In this section we evaluate our synset representations on the Wikipedia sense clustering task. For a fair comparison with respect to the BabelNet-based comparison systems that use the Wikipedia corpus for training, in this experiment we report the results of our model trained on the Wikipedia corpus and using BabelNet as lexical resource only. For the evaluation we consider the two Wikipedia sense clustering datasets (500-pair and SemEval) created by Dandala et al. (2013). In these datasets sense clustering is viewed as a binary classification task in which, given a pair of Wikipedia pages, the system has to decide whether to cluster them into a single instance or not. To this end, we use our synset embeddings and cluster Wikipedia pages[11] together if their similarity exceeds a threshold γ. In order to set the optimal value of γ, we follow Dandala et al. (2013) and use the first 500-pairs sense clustering dataset for tuning. We set the threshold γ to 0.35, which is the value 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 threshold for NASARI (0.7) and SensEmbed (0.3) comparison systems.

Finally, we evaluate our approach on the SemEval sense clustering test set. This test set consists of 925 pairs which were obtained from a set of highly ambiguous words gathered from past SemEval tasks. For comparison, we also include the supervised approach of Dandala et al. (2013) based on a multi-feature Support Vector Machine classifier trained on an automatically-labeled dataset of the English Wikipedia (Mono-SVM) and Wikipedia in four different languages (Multi-SVM). As naive baseline we include the system which would cluster all given pairs.

Table 4.4 shows the F-Measure and accuracy results on the SemEval sense clustering dataset. SW2V outperforms all comparison systems according to both measures, including the sense representations of NASARI and SensEmbed using the same setup and the same

[11] Since Wikipedia is a resource included in BabelNet, our synset representations are expandable to Wikipedia pages.

Model	Accuracy	F-Measure
Baseline	17.5	29.8
SensEmbed	82.7	40.3
NASARI	87.0	62.5
Multi-SVM	85.5	-
Mono-SVM	83.5	-
SW2V	**87.8**	**63.9**

Table 4.4. Accuracy and F-Measure percentages of different systems on the SemEval Wikipedia sense clustering dataset.

underlying lexical resource. This confirms the capability of our system to accurately capture the semantics of word senses on this sense-specific task.

4.5.3 Word and sense interconnectivity

In the previous experiments we evaluated the effectiveness of the sense embeddings. In contrast, this experiment aims at testing the interconnectivity between word and sense embeddings in the vector space. As explained in Section 2, there have been previous approaches building a shared space of word and sense embeddings (Chen et al., 2014; Rothe and Schütze, 2015), 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 (Navigli, 2009; 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

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

Table 4.5. F-Measure percentage of different MCS strategies on the SemEval-2007 and SemEval-2013 WSD datasets.

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: 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.

Table 4.5 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 4.6 shows the closest word and sense[12] 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

[12] Following Navigli (2009), *word*$_n^p$ is the n^{th} sense of *word* with part of speech p (using WordNet 3.0).

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

Table 4.6. Ten closest word and sense embeddings to the senses *company*^2_n (military unit) and *school*^7_n (group of fish).

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*^2_n and *school*^7_n senses.

4.6 Conclusions

In this chapter we propose 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.

Chapter 5

Embeddings for Word Sense Disambiguation

"... the complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously"

John Rupert Firth, 1935

Recent years have seen a dramatic growth in the popularity of word embeddings mainly owing to their ability to capture semantic information from massive amounts of textual content. As a result, many tasks in NLP have tried to take advantage of the potential of these distributional models. While in the previous chapterw we show how Word Sense Disambiguation can be leveraged to learn better embeddings, going from words to senses, in this chapter we study how word embeddings can be leveraged to build better models for WSD. We propose different methods through which word embeddings can be leveraged for performing WSD, and perform a deep analysis of how different parameters affect performance. We show how a WSD system that makes use of word embeddings alone, if designed properly, can provide significant performance improvement over a state-of-the-art WSD system that incorporates several standard WSD features.

5.1 Introduction

Embeddings represent words, or concepts in a low-dimensional continuous space. These vectors capture useful syntactic and semantic information, such as regularities in language, where relationships are characterized by a relation-specific vector offset. The ability of embeddings to capture knowledge has been exploited in several tasks, such as Machine Translation (Mikolov et al., 2013b, MT), Sentiment Analysis (Socher et al., 2013), Word Sense Disambiguation (Chen et al., 2014, WSD) and Language Understanding (Mesnil et al., 2013). Supervised WSD is based on the hypothesis that contextual information provides a good approximation to word meaning, as suggested by Miller and Charles Miller and Charles (1991): semantically similar words tend to have similar contextual distributions.

Recently, there have been efforts on leveraging embeddings for improving supervised WSD systems. Taghipour and Ng (2015b) showed that the performance of conventional supervised WSD systems can be increased by taking advantage of embeddings as new features. In the same direction, Rothe and Schütze (2015) trained embeddings by mixing words, lexemes and synsets, and introducing a set of features based on calculations on the resulting representations. However, none of these techniques takes full advantage of the semantic information contained in embeddings. As a result, they generally fail in providing substantial improvements in WSD performance.

In this chapter, we provide a study of different techniques for taking advantage of the combination of embeddings with standard WSD features. We also propose an effective approach for leveraging embeddings in WSD, and show that this can provide significant improvement on multiple standard benchmarks.

5.2 Word Embeddings

An embedding is a representation of a topological object, such as a manifold, graph, or field, in a certain space in such a way that its connectivity or algebraic properties are preserved (Insall et al., 2015). Presented originally by Bengio et al. (2003), word embeddings aim at representing, i.e., embedding, the ideal semantic space of words in a real-valued continuous vector space. In contrast to traditional distributional techniques, such as Latent Semantic Analysis (LSA) (Landauer and Dutnais, 1997) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Bengio et al. (2003) designed a feed-forward neural network capable of predicting a word given the words preceding (i.e., leading up to) that word. Collobert and Weston (2008) presented a much deeper model consisting of several layers for feature extraction, with the objective of building a general architecture for NLP tasks. A major breakthrough occurred when Mikolov et al. (2013b) put forward an efficient algorithm for training embeddings, known as word2vec. A similar model to word2vec was presented by Pennington et al. (2014), GloVe, but instead of using latent features for representing words, it makes an explicit representation produced from statistical calculation on word countings.

In the following section we discuss how embeddings can be integrated into an important lexical semantic task, i.e., Word Sense Disambiguation.

5.3 Word Sense Disambiguation

Natural language is inherently ambiguous. Most commonly-used words have several meanings. In order to identify the intended meaning of a word one has to analyze the context in which it appears by directly exploiting information from raw texts. The task of automatically assigning predefined meanings to words in contexts, known as Word Sense Disambiguation, is a fundamental task in computational lexical semantics, as we mentioned earlier in this Thesis (cf. Section 1.3).

5.3.1 Standard WSD features

As was analyzed by Lee and Ng (2002), conventional WSD systems usually make use of a fixed set of features to model the context of a word. The first feature is based on the words in the surroundings of the target word. The feature usually represents the local context as a binary array, where each position represents the occurrence of a particular word. Part-of-speech (POS) tags of the neighboring words have also been used extensively as a WSD feature. Local collocations represent another standard feature that captures the ordered sequences of words which tend to appear around the target word (Firth, 1957). Though not very popular, syntactic relations have also been studied as a possible feature in WSD (Stetina et al., 1998).

More sophisticated features have also been studied. Examples are distributional semantic models, such as Latent Semantic Analysis (Van de Cruys and Apidianaki, 2011) and Latent Dirichlet Allocation (Cai et al., 2007). Inasmuch as they are the dominant distributional semantic model, word embeddings have also been applied as features to WSD systems. In this chapter we study different methods through which word embeddings can be used as WSD features.

5.3.2 Word Embeddings as WSD features

Word embeddings have become a prominent technique in distributional semantics. These methods leverage neural networks in order to model the contexts in which a word is expected to appear. Thanks to their ability in efficiently learning the semantics of words, word embeddings have been applied to a wide range of NLP applications. Several studies have also investigated their integration into the Word Sense Disambiguation setting. These include the works of Zhong and Ng (2010); Taghipour and Ng (2015b); Rothe and Schütze (2015); Chen et al. (2014), which leverage embeddings for supervised (the former three) and knowledge-based (the latter) WSD. However, to our knowledge, no previous work has investigated methods for integrating word embeddings in WSD and the role that different training parameters can play. In this chapter, we put forward a framework for a comprehensive evaluation of different methods of leveraging word embeddings as WSD features in a supervised WSD system. We provide an analysis of the impact of different parameters in the training of embeddings on the WSD performance. We consider four different strategies for integrating a pre-trained word embedding in a supervised WSD system, discussed in what follows.

Concatenation

Concatenation is our first strategy, which is inspired by the model of Bengio et al. (2003). This method consists of concatenating the vectors of the words surrounding a target word into a larger vector that has a size equal to the aggregated dimensions of all the individual embeddings. Let w_{ij} be the weight associated with the i^{th} dimension of the vector of the j^{th} word in the sentence, let D be the dimensionality of this vector, and W be the window size which is defined as the number of words on a single side. We are interested in representing the context of the I^{th} word in the sentence. The i^{th} dimension of the concatenation feature vector, which has a size of $2WD$, is computed as follows:

$$e_i = \begin{cases} w_{i \bmod D,\, I-W+\lfloor \frac{i}{D} \rfloor} & \text{if } \lfloor \frac{i}{D} \rfloor < W \\ w_{i \bmod D,\, I-W+1+\lfloor \frac{i}{D} \rfloor} & \text{otherwise} \end{cases} \tag{5.1}$$

where *mod* is the modulo operation, i.e., the remainder after division.

Average

As its name indicates, the average strategy computes the centroid of the embeddings of all the surrounding words. The formula divides each dimension by $2W$ since the number of context words is twice the window size:

$$e_i = \sum_{\substack{j=I-W \\ j \neq I}}^{I+W} \frac{w_{ij}}{2W} \tag{5.2}$$

Fractional decay

Our third strategy for constructing a feature vector on the basis of the context word embeddings is inspired by the way word2vec combines the words in the context. Here, the importance of a word for our representation is assumed to be inversely proportional to its distance from the target word. Hence, surrounding words are weighted based on their distance from the target word:

$$e_i = \sum_{\substack{j=I-W \\ j \neq I}}^{I+W} w_{ij} \frac{W - |I-j|}{W} \tag{5.3}$$

Exponential decay

Exponential decay functions similarly to the fractional decay, which gives more importance to the close context, but in this case the weighting in the former is performed exponentially:

$$e_i = \sum_{\substack{j=I-W \\ j \neq I}}^{I+W} w_{ij} (1-\alpha)^{|I-j|-1} \tag{5.4}$$

where $\alpha = 1 - 0.1^{(W-1)^{-1}}$ is the decay parameter. We choose the parameter in such a way that the immediate surrounding words contribute 10 times more than the last words on both sides of the window.

5.4 WSD Framework

In the following we describe two models for performing supervised and unsupervised Word Sense Disambiguation.

5.4.1 Supervised WSD System

With this models our goal is to experiment with state-of-the-art conventional supervised WSD system and a varied set of word embedding techniques. We selected IMS (Zhong and Ng, 2010) as our underlying framework for supervised WSD. IMS provides an extensible and flexible platform for supervised WSD by allowing the verification of different WSD features and classification techniques. By default, IMS makes use of three sets of features: (1) POS tags of the surrounding words, with a window of three words on each side, restricted by the sentence boundary, (2) the set of words that appear in the context of the target word after stopword removal, and (3) local collocations which consist of 11 features around the target word. IMS uses a linear support vector machine (SVM) as its classifier. We take the real-valued word embeddings as new features of IMS and introduce them into the system without performing any further modifications.

Embedding Features. We carried out experiments with three different embeddings:

- **word2vec** Mikolov et al. (2013b): We used the word2vec toolkit to learn 400 dimensional vectors on the September-2014 dump of the English Wikipedia which comprises around three billion tokens. We chose the Skip-gram architecture with the negative sampling set to 10. The sub-sampling of frequent words was set to 10^{-3} and the window size to 10 words.

- **C&W** Collobert and Weston (2008): These 50 dimensional embeddings were learnt using a neural network model, consisting of several layers for feature extraction. The vectors were trained on a subset of the English Wikipedia.[1]

- **Retrofitting**: Finally, we used the approach of Faruqui et al. (2015) to retrofit our word2vec vectors. We used the Paraphrase Database (Ganitkevitch et al., 2013, PPDB) as external knowledge base for retrofitting and set the number of iterations to 10.

5.4.2 Unsupervised WSD System

We present also an alternative model for performing Substitution-based WSD. Our model, namely LexSubEmbed, creates a vector which represent the meaning of the word in context by making a weighed sum of its surrounding. The strategy for constructing a feature vector on the basis of the context word embeddings is inspired in the way in which word2vec combines the surroundings words of the context. Here, the contribution of a word in our representation is assumed to be inversely proportional to its distance from the target word. Hence, surrounding words are weighted based on their distance from the target word. In our case, the weighting in the former is performed utilizing the **Exponential decay** strategy (cf. 5.4). For as embeddings, we use the **Retrofitting** configuration, which was our best model using only embeddings as features. (cf. 5.5.1).

5.5 Experiments

We evaluated the performance of our embedding-based WSD system on three standard tasks: lexical sample, all-words WSD and lexical substitution. In all the experiments in this section we used the exponential decay strategy (cf. Section 5.3.2) and a window size of ten words on each side of the target word.

5.5.1 Lexical Sample WSD Experiment

The lexical sample WSD tasks provide training datasets in which different occurrences of a small set of words are sense annotated. The goal is for a WSD system to analyze the contexts of the individual senses of these words and to capture clues that can be used for distinguishing different senses of a word from each other at the test phase.

[1] `http://ronan.collobert.com/senna/`

Task	**Training**			**Test**		
	noun	**verb**	**adjective**	**noun**	**verb**	**adjective**
Senseval-2 (SE2)	4851	3566	755	1740	1806	375
Senseval-3 (SE3)	3593	3953	314	1807	1978	159
SemEval-07 (SE7)	13287	8987	–	2559	2292	–

Table 5.1. The number of sentences per part of speech in the datasets of the English lexical sample tasks we considered for our experiments.

Datasets. As our benchmark for the lexical sample WSD, we chose the Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Mihalcea et al., 2004), and SemEval-2007 (Pradhan et al., 2007) English Lexical Sample WSD tasks. The former two cover nouns, verbs and adjectives in their datasets whereas the latter task focuses on nouns and verbs only. Table 5.1 shows the number of sentences per part of speech for the training and test datasets of each of these tasks.

Comparison systems. In addition to the vanilla IMS system in its default setting we compared our system against two recent approaches that also modify the IMS system so that it can benefit from the additional knowledge derived from word embeddings for improved WSD performance: (1) the system of Taghipour and Ng (2015b), which combines word embeddings of Collobert and Weston (2008) using the concatenation strategy (cf. Section 5.3.2) and introduces the combined embeddings as a new feature in addition to the standard WSD features in IMS; and (2) AutoExtend (Rothe and Schütze, 2015), which constructs a whole new set of features based on vectors made from words, senses and synsets of WordNet and incorporates them in IMS.

Lexical sample WSD results

Table 5.2 shows the F1 performance of the different systems on the three lexical sample datasets. As can be seen, the IMS + word2vec system improves over all comparison systems including those that combine standard WSD and embedding features (i.e., the system of Taghipour and Ng, and AutoExtend) across all the datasets. This shows that our proposed strategy for introducing word embeddings into the IMS system on the basis of exponential

System	**SE2**	**SE3**	**SE7**
IMS (Zhong and Ng, 2010)	65.3	72.9	87.9
Taghipour and Ng (2015b)	66.2	73.4	–
AutoExtend (Rothe and Schütze, 2015)	66.5	73.6	–
IMS + C&W	64.3	70.1	88.0
IMS + word2vec	**69.9**	**75.2**	**89.4**
IMS + Retrofitting	65.9	72.8	88.3
C&W feature only	55.0	61.6	83.4
word2vec feature only	65.6	69.4	87.0
Retrofitting feature only	67.2	72.7	88.0

Table 5.2. F1 performance on the three English lexical sample datasets. IMS + X denotes the improved IMS system when the X set of word representations were used as additional features. We also show in the last three rows the results for the IMS system when word representations were used as the only features.

decay was beneficial. In the last three rows of the table, we also report the performance of the WSD systems that leverage only word embeddings as their features and do not incorporate any standard WSD feature. It can be seen that word embeddings, in isolation, provide competitive performance, which proves their capability in obtaining the information captured by standard WSD features. Among different embeddings, the retrofitted vectors provide the best performance when used in isolation.

5.5.2 All-Words WSD Experiments

The goal in this task is to disambiguate all the content words in a given text. In order to learn models for disambiguating a large set of content words, a high-coverage sense-annotated corpus is required. Since all-words tasks do not usually provide any training data, the challenge here is not only to learn accurate disambiguation models from the training data, as is the case in the lexical sample task, but also to gather high-coverage training data and to learn disambiguation models for as many words as possible.

Training corpus. As our training corpus we opted for two available resources: SemCor and OMSTI. SemCor (Miller et al., 1994) is a manually sense-tagged corpus created by the WordNet project team at Princeton University. The dataset is a subset of the English Brown Corpus and comprises around 360,000 words, providing annotations for more than 200K content words.[2] OMSTI[3] (Taghipour and Ng, 2015a) (One Million Sense-Tagged for Word Sense Disambiguation and Induction) was constructed based on the DSO corpus (Ng and Lee, 1996) and provides annotations for around 42K different nouns, verbs, adjectives, and adverbs.

Datasets. As benchmark for this experiment, we considered the Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), and SemEval-2007 (Pradhan et al., 2007) English all-words tasks. There are 2474, 2041, and 465 words for which at least one of the occurrences has been sense annotated in the Senseval-2, Senseval-3 and SemEval-2007 datasets, respectively.

Experimental setup. Similarly to the lexical sample experiment, in the all-words setting we used the exponential decay strategy (cf. Section. 5.4.1) in order to incorporate word embeddings as new features in IMS. For this experiment, we only report the results for the best-performing word embeddings in the lexical sample experiment, i.e., word2vec (see Table 5.2).

Comparison systems. We benchmarked the performance of our system against five other systems. Similarly to our lexical sample experiment, we compared against the vanilla IMS system and the work of Taghipour and Ng (2015b). In addition, we performed experiments on the nouns subsets of the datasets in order to be able to provide comparisons against two other WSD approaches: Babelfy (Moro et al., 2014) and Muffin (Camacho-Collados et al., 2015a). Babelfy is a multilingual knowledge-based WSD and Entity Linking algorithm based on the semantic network of BabelNet. Muffin is a multilingual sense representation technique that combines the structural knowledge derived from semantic networks with the distributional statistics obtained from text corpora. The system uses sense-based representations for

[2] We used automatic mappings to WordNet 3.0 provided in `web.eecs.umich.edu/∼mihalcea/downloads.html`.

[3] `www.comp.nus.edu.sg/~nlp/corpora.html`

performing WSD. (Camacho-Collados et al., 2015a) also proposed a hybrid system that averages the disambiguation scores of IMS with theirs (shown as “Muffin + IMS" in our tables). We also report the results for UKB w2w (Agirre and Soroa, 2009), another knowledge-based WSD approach based on Personalized PageRank (Haveliwala, 2002, PPR). Finally, we also carried out experiments with the pre-trained models[4] that are provided with the IMS toolkit, as well as IMS trained on our two training corpora, i.e., SemCor and OMSTI.

All-words WSD results

Tables 5.3 and 5.4 list the performance of different systems on, respectively, the whole and the noun-subset datasets of the three all-words WSD tasks. Similarly to our lexical sample experiment, the IMS + word2vec system provided the best performance across datasets and benchmarks. The coupling of word2vec embeddings to the IMS system proved to be consistently helpful. Among the two training corpora, as expected, OMSTI provided a better performance owing to its considerably larger size and higher coverage. Another point to be noted here is the difference between results of the IMS with the pre-trained models and those trained on the OMSTI corpus. Since we used the same system configuration across the two runs, we conclude that the OMSTI corpus is either substantially smaller or less representative than the corpus used by Zhong and Ng (2010) for building the pre-trained models of IMS. Despite this fact, the IMS + word2vec system can consistently improve the performance of IMS (pre-trained models) across the three datasets. This shows that a proper introduction of word embeddings into a supervised WSD system can compensate the negative effect of using lower quality training data.

5.5.3 Lexical Substitution Experiment

Finding alternative words that can occur in given contexts constitutes the main objective of the lexical substitution task. We chose the dataset used in the original SemEval-2007 shared task (McCarthy and Navigli, 2007), which consists of 201 words manually chosen to exhibit polysemy, with 10 sentences per target. For a given target in a particular context, five annotators were asked to propose up to 3 substitutes. As all our experiments are

[4] www.comp.nus.edu.sg/~nlp/sw/models.tar.gz

System	SE2	SE3	SE7
MFS baseline	60.1	62.3	51.4
IMS (Zhong and Ng, 2010)	68.2	67.6	58.3
Taghipour and Ng (2015b)	–	**68.2**	–
IMS (pre-trained models)	67.7	67.5	58.0
IMS (SemCor)	62.5	65.0	56.5
IMS (OMSTI)	67.0	66.4	57.6
IMS + word2vec (SemCor)	63.4	65.3	57.8
IMS + word2vec (OMSTI)	**68.3**	**68.2**	**59.1**

Table 5.3. F1 performance on different English all-words WSD datasets.

System	SE2	SE3	SE7
MFS baseline	71.6	70.3	65.8
Babelfy	–	68.3	62.7
Muffin	–	–	66.0
Muffin + IMS	–	–	68.5
UBK w2w	–	65.3	56.0
IMS (pre-trained models)	77.5	74.0	66.5
IMS (SemCor)	73.0	70.8	64.2
IMS (OMSTI)	76.6	73.3	67.7
IMS + word2vec (SemCor)	74.2	70.1	68.6
IMS + word2vec (OMSTI)	**77.7**	**74.1**	**71.5**

Table 5.4. F1 performance in the nouns subsets of different all-words WSD datasets.

unsupervised, we always evaluate over the entire data set, rather than the original held-out test set. Table 5.5 shows the number of sentences per part of speech.

Comparison systems. We compared our approach with other system, some of which where participants in the SemEval task. KU (Yuret, 2007) introduced a WSD model which uses a statistical language model combined with substitutes candidates from WordNet and the Roget Thesaurus (Thesaurus.com, 2007). University of North Texas SUBFINDER (Hassan et al., 2007, UNT), a model specially created for this task which extract possible substitutes from a variety of knowledge source and provided a list of candidates based on a weighted combination of a set of common ranking methods as MFS, Language Modelling, Latent Semantic Analysis (LSA), Information Retrieval (IR), and Word Sense Disambiguation (WSD). IRST2 is one of two models introduced by Giuliano et al. (2007) for the same task. The complete names IRST1-lsa and IRST2-syn exploit the fact that lexical substitution can be seen as a subtask of lexical entailment. To this end the author use two rankings for choosing the candidates extracted from WordNet and Oxford dictionary: Domain Proximity, based on LSA, and Syntagmatic Coherence, by querying target sentence in a large corpus. (Van de Cruys et al., 2011, NMF) presented a new method for computating word meaning in context which uses a factorization model in which words, together with their window-based context words and their dependency relations, are linked to latent dimensions. Next, Melamud et al. (2015b) utilized the context embeddings from Skip-gram, those which are normally discarded, and used them for measuring ranking the substitutes and in Melamud et al. (2015a) they extended the approach by proposing a new distributional model for representing word meaning in context, based on this context representation. Finally, Hintz and Biemann (2016) introduced an approach for effective lexical for lexical substitution able to perform transfer learning across languages, in particular English, Italian and German.

Lexical substitution results

Table 5.6 shows the performance of the different systems on the SemEval-2007 task for Lexical Substitution. We include the four measurements of the task, best, best-mode, oot and oot-mode. best measures the precision as first answer, while oot measures the ordering of the ten first guesses. The *mode* takes the first (or out of ten) guess and compare against

Task	**Test**				
	noun	**adjective**	**verb**	**adverb**	Total
SemEval-07 Task 10	500	470	440	300	2010

Table 5.5. The number of sentences per part of speech in the Lexical Substitution task .

System	Cand.	best	best-mode	oot	oot-mode
KU (Yuret, 2007)	WN	12.90	20.65	46.15	61.30
UNT (Hassan et al., 2007)	WN	12.77	20.73	49.19	66.26
IRST2 (Giuliano et al., 2007)	U	6.94	20.33	68.96	58.54
NMF (Van de Cruys et al., 2011)	U	8.96	-	29.26	-
DelexFeat (Szarvas et al., 2013)	Gold	15.94	-	-	-
Simple (Melamud et al., 2015b)	U	8.14	13.41	27.42	39.11
P^{in}_{1000} (Melamud et al., 2015a)	U	12.72	21.71	36.37	52.03
Hintz and Biemann (2016)	Gold	16.63	-	48.16	-
LexSubEmbed	WN	13.99	23.90	33.90	43.01
	Gold	**17.88**	**26.63**	**72.31**	**83.02**

Table 5.6. Performance of different systems for the four measurements included on the SemEval-2007 English Lexical Substitution Task.

the mode of the annotators. Our approach, LexSubEmbed, uses two strategies for choosing the candidates, both of which outperforms models using the same resource both on best and oot results.

5.6 Analysis

We carried out a series of experiments in order to check the impact of different system parameters on the final WSD performance. We were particularly interested in observing the role that various training parameters of embeddings as well as WSD features have in the WSD performance. We used the Senseval-2 English Lexical Sample task as our benchmark for this analysis.

5.6.1 The effect of different parameters

Table 5.7 shows F1 performance of different configurations of our system on the task's dataset. We studied five different parameters: the type (i.e., w2v or Retrofitting) and dimensionality (200, 400, or 800) of the embeddings, combination strategy (concatenation, average, fractional or exponential decay), window size (5, 10, 20 and words), and WSD features (collocations, POS tags, surrounding words, all of these or none). All the embeddings in this experiment were trained on the same training data and, unless specified, with the same configuration as described in Section 5.4.1. As baseline we show in the table the performance of the vanilla WSD system, i.e., IMS. For better readability, we report the differences between the performances of our system and the baseline.

Collocations				✓							✓			✓				
POS							✓			✓			✓					
Surroundings				✓			✓						✓					
Dimensionality			200	400	800	200	400	800	200	400	800	200	400	800	200	400	800
System	Strategy	Window															
IMS			62.4			63.7			62.0			65.2			–		
+ w2v	Con	5	+0.1	+0.4	+0.1	-0.1	+0.3	+0.2	+0.1	+0.5	+0.1	-0.2	+0.1	+0.1	46.9	48.7	44.2
		10	-0.1	+0.5	+0.3	-0.1	+0.5	0.0	+0.6	+1.0	+0.5	-0.1	+0.1	-0.1	48.6	51.1	49.7
		20	-0.2	+0.4	—	-0.3	+0.3	—	+0.7	+1.5	—	-0.5	+0.4	—	52.5	54.1	—
+ w2v	Avg	5	+0.8	+1.0	+1.0	+1.3	+1.3	+1.4	+3.9	+4.2	+4.1	+1.7	+1.4	+1.6	58.3	59.9	61.3
		10	+0.8	+0.9	+0.9	+0.6	+0.7	+0.8	+3.6	+3.7	+3.9	+0.6	+0.6	+0.7	63.7	64.1	64.7
		20	+0.3	+0.3	+0.3	+0.5	+0.3	+0.4	+2.4	+2.3	+2.3	+0.2	+0.2	+0.2	62.7	63.1	63.5
+ w2v	Frac	5	+3.9	+4.9	+5.2	+4.2	+4.6	+5.3	+6.3	+6.6	+6.8	+3.0	+3.6	+3.8	61.2	63.1	64.8
		10	+4.9	+5.8	+5.7	+4.6	+5.2	+5.1	+5.9	+7.0	+7.4	+3.6	+4.3	+4.0	61.3	63.8	65.2
		20	+4.4	+4.5	+4.7	+3.7	+4.0	+4.3	+4.8	+6.1	+5.4	+3.2	+3.3	+3.4	61.2	63.4	63.9
+ w2v	Exp	5	+4.1	+5.0	+5.2	+4.1	+4.7	+5.0	+6.1	+6.1	+6.4	+2.9	+3.5	+3.7	62.3	64.7	64.9
		10	+5.4	**+6.6**	+6.4	+4.9	+5.8	**+6.0**	+7.2	+7.7	**+8.2**	+4.1	**+4.7**	+4.6	63.2	65.6	66.9
		20	+5.2	+5.6	+5.9	+4.4	+5.1	+4.9	+6.1	+7.0	+6.8	+3.9	+4.3	+4.2	61.9	64.4	65.2
+ Ret	Con	5	-0.1	-0.1	-0.1	-0.1	-0.1	0.0	+0.1	+0.1	-0.1	-0.1	+0.1	+0.1	50.7	53.5	50.9
		10	+0.1	0.0	0.0	-0.3	0.0	0.0	+0.1	+0.2	+0.1	0.0	0.0	0.0	52.1	54.2	53.4
		20	0.0	0.0	—	-0.2	0.0	—	+0.7	+0.3	—	0.0	-0.1	—	53.7	54.8	—
+ Ret	Avg	5	+0.1	0.0	-0.1	+0.1	0.0	-0.1	+0.8	+0.8	+0.7	+0.1	0.0	+0.1	60.7	60.3	60.5
		10	-0.2	-0.1	0.0	-0.2	-0.3	0.0	+0.7	+0.7	+0.5	0.0	+0.1	+0.1	58.9	58.4	58.2
		20	-0.1	+0.1	+0.1	-0.2	-0.2	-0.2	+0.5	+0.4	+0.4	0.0	0.0	0.0	56.5	56.0	55.5
+ Ret	Frac	5	+1.4	+1.3	+1.2	+1.2	+1.0	+0.9	+3.3	+3.1	+2.9	+0.5	+0.3	+0.3	66.5	67.3	67.7
		10	+1.7	+1.4	+1.2	+1.5	+1.4	+1.2	+5.2	+4.7	+4.5	+0.7	+0.8	+0.6	64.4	66.2	66.1
		20	+2.2	+2.2	+1.8	+2.2	+1.8	+2.0	+6.7	+6.4	+5.9	+1.3	+1.2	+1.0	64.0	64.2	64.7
+ Ret	Exp	5	+1.1	+1.1	+1.1	+0.8	+0.8	+0.7	+2.7	+2.6	+2.2	+0.3	+0.3	+0.3	66.8	67.7	**68.0**
		10	+1.5	+1.3	+1.0	+1.2	+1.1	+1.0	+4.4	+4.2	+3.8	+0.7	+0.7	+0.3	65.9	67.2	67.5
		20	+1.8	+1.7	+1.5	+1.7	+1.5	+1.5	+6.3	+5.9	+5.4	+1.1	+0.8	+0.7	65.1	65.8	66.5

Table 5.7. F1 performance of different models on the Senseval-2 English Lexical Sample task. We show results for varied dimensionality (200, 400, and 800), window size (5, 10 and 20 words) and combination strategy, i.e., Concatenation (Con), Averaging (Avg), Fractional decay (Frac), and Exponential decay (Exp). To make the table easier to read, we highlight each cell according to the relative performance gain in comparison to the IMS baseline (top row in the table).

We observe that the addition of word2vec word embeddings to IMS (+w2v in the table) was beneficial in all settings. Among combination strategies, concatenation and average produced the smallest gain and did not benefit from embeddings of higher dimensionality. However, the other two strategies, i.e., fractional and exponential decay, showed improved performance with the increase in the size of the employed embeddings, irrespective of the WSD features. The window size showed a peak of performance when 10 words were taken in the case of standard word embeddings. For retrofitting, a larger window seems to have been beneficial, except when no standard WSD features were taken. Another point to note here is that, among the three WSD features, POS proved to be the most effective one while due to the nature of the embeddings, the exclusion of the Surroundings features in addition to the inclusion of the embeddings was largely beneficial in all the configurations. Furthermore, we found that the best configurations for this task were the ones that excluded Surroundings, and included w2v embeddings with a window of 10 and 800 dimensions with exponential decay strategy (70.2% of F1 performance) as well as the configuration used in our experiments, with all the standard features, and w2v embeddings with 400 dimensions, a window of 10 and exponential decay strategy (69.9% of F1 performance).

The retrofitted embeddings provided lower performance improvement when added on top of standard WSD features. However, when they were used in isolation (shown in the right-most column), the retrofitted embeddings interestingly provided the best performance, improving the vanilla WSD system with standard features by 2.8 percentage points (window size 5, dimensionality 800). In fact, the standard features had a destructive role in this setting as the overall performance was reduced when they were combined with the retrofitted embeddings. Finally, we point out the missing values in the configuration with 800 dimensions and a window size of 20. Due to the nature of the concatenation strategy, this configuration greatly increased the number of features from embeddings only, reaching 32000 (800 x 2 x 20) features. Not only was the concatenation strategy unable to take advantage of the increased dimensionality, but also it was not able to scale.

These results show that a state-of-the-art supervised WSD system can be constructed without incorporating any of the conventional WSD features, which in turn demonstrates the potential of retrofitted word embeddings for WSD. This finding is interesting, because it provides the basis for further studies on how synonymy-based semantic knowledge introduced by retrofitting might play a role in effective WSD, and how retrofitting might be optimized for improved WSD. Indeed, such studies may provide the basis for re-designing the standard WSD features.

5.6.2 Comparison of embedding types

We were also interested in comparing different types of embeddings in our WSD framework. We tested for seven sets of embeddings with different dimensionalities and learning techniques: word2vec embeddings trained on Wikipedia, with the Skip-gram model for dimensionalities 50, 300 and 500 (for comparison reasons) and CBOW with 300 dimensions, word2vec trained on the Google News corpus with 300 dimensions and the Skip-gram model, the 300 dimensional embeddings of GloVe, and the 50 dimensional C&W embeddings. Additionally we include experiments on a non-embedding model, a PMI-SVD vector space model trained by Baroni et al. (2014).

Table 5.8 lists the performance of our system with different word representations in vector space on the Senseval-2 English Lexical Sample task. The results corroborate the findings of Levy et al. (2015) that Skip-gram is more efficient in capturing the semantics than CBOW and GloVe. Additionally, the use of embeddings with decay fares well, independently of the type of embedding. The only exception is the C&W embeddings, for which the average strategy works best. We attribute this behavior to the nature of these embeddings, rather than to their dimensionality. This is shown in our comparison against the 50-dimensional Skip-gram embeddings trained on the Wikipedia corpus (bottom of Table 5.8), which performs well with both decay strategies, outperforming C&W embeddings.

5.7 Conclusions

In this chapter we study different ways of taking advantage of the semantic knowledge of word embeddings for performing WSD. We carried out a deep analysis of different parameters and strategies across several WSD tasks. We draw three main findings. First, word embeddings can be used as features to improve both supervised and unsupervised WSD systems. Second, utilizing embeddings on the basis of an exponential decay strategy proves to be more consistent in producing high performance than the other conventional strategies, such as vector concatenation and centroid. Third, the retrofitted embeddings that

Word representations	**Dim.**	**Combination strategy**			
		Concatenation	Average	Fractional	Exponential
Skip-gram - GoogleNews	300	65.5	65.5	69.4	**69.6**
GloVe	300	61.7	66.3	66.7	**68.3**
CBOW - Wiki	300	65.1	65.4	**68.9**	68.8
Skip-gram - Wiki	300	65.2	65.6	68.9	**69.7**
PMI - SVD - Wiki	500	65.5	65.3	**67.3**	66.8
Skip-gram - Wiki	500	65.1	65.6	69.1	**69.9**
Collobert & Weston	50	58.6	**67.3**	62.9	64.3
Skip-gram - Wiki	50	65.0	65.7	68.3	**68.6**

Table 5.8. F1 percentage performance on the Senseval-2 English Lexical Sample dataset with different word representations models, vector dimensionalities (Dim.) and combination strategies.

take advantage of the knowledge derived from semi-structured resources, when used as the only feature for WSD can outperform state-of-the-art supervised models which use standard WSD features. However, the best performance is obtained when standard WSD features are augmented with the additional knowledge from word2vec vectors on the basis of a decay function strategy. We release at `https://github.com/iiacobac/ims_wsd_emb` all the codes and resources used in our experiments in order to provide a framework for research on the evaluation of new VSM models in the WSD framework.

Chapter 6

SENSEMBED+: A "Meaningful" Vector Space Model

> "Purpose requires an understanding of intent. Which means we have to find out if they make conscious choices or if their motivation is so instinctive they don't understand a "why" question at all, We need to have enough vocabulary with them so we understand their answer."
>
> *ARRIVAL Movie Screenplay*

Current mainstream Natural Language Processing approaches are enabled by continuous vector representations of words. However, the most popular representations conflate the various senses of polysemous words in a single vector, which limits their use to predominant meanings. To address this issue, we present SENSEMBED+, a joint model for effective word, sense and synset representation in a common semantic vector space, which brings together the knowledge available in existing lexical-semantic resources and the distributional information obtained from large amounts of raw corpora. We evaluate our model on various tasks achieving state-of-the-art performance in several standard benchmarks and applications, including word, sense, and relational similarity, and Word Sense Disambiguation.

6.1 Introduction

In recent years, *word embeddings* have represented one of the most promising breakthroughs in Natural Language Processing (Goldberg, 2017). In a nutshell, they are computational models where each word is represented by a vector in a low-dimensional space, typically computed by a neural network trained on a large text corpus (Collobert and Weston, 2008; Mnih and Hinton, 2009; Turian et al., 2010; Mikolov et al., 2013b; Pennington et al., 2014). Word vectors are functions of the contexts in which the words are observed in the corpus (hence the naming *embedding*), therefore the resulting models instantiate the principle of *distributional semantics*: words that share similar contexts are related in meaning (Harris, 1954; Firth, 1957). In the vector space, this translates to the interesting property of semantically related words being represented by vectors with a short distance to each other. Moreover, semantic relations such as hypernymy and analogy can be modeled in a word vector space as geometric transformations (Baroni and Zamparelli, 2010; Wang et al., 2014b). As such, word embeddings have been successfully used to tackle a large number of NLP tasks involving lexical semantics and the relations between the meaning of words, such as semantic word similarity, word sense disambiguation, text classification, and so on.

Despite their success, word embedding models are not devoid of limitations. Word embeddings are solely trained with large amounts of raw data, neglecting existing semantic resources such as knowledge bases, ontologies, thesauri and semantic networks. While in a pure distributional approach a pair of synonyms, for instance the verbs *buy* and *purchase*, is inferred based on their vectorial proximity induced by their mutual word cooccurrences, in lexical-semantic resources like WordNet (Miller, 1995) this information is structured and verified manually, with synonymous words grouped into unordered sets called *synsets*. For instance, the word senses buy_v^1 and $purchase_v^1$ belong to the same synset[1]. Another important issue is that distributional approaches conflate the senses of a word into a single vector, therefore hindering a correct representation of polysemy. This is an issue for most tasks dealing with semantic relatedness, as, for instance, the closest words in a word

[1] We follow Navigli (2009) and show the n^{th} sense of the *word* with part of speech p as $word_p^n$.

embedding model to *rock* could include words such as *stone* and *earth*, but also *music* and *band*.

To overcome such limitations, several embedding models have been recently proposed. Neelakantan et al. (2014) and Tian et al. (2014) propose purely distributional approaches where word senses are induced by clustering the word vectors to learn multi-prototype embeddings. Yu and Dredze (2014) and Faruqui et al. (2015) include semantic knowledge from existing knowledge resources to improve the word embeddings. Hassan and Mihalcea (2011), Wu and Giles (2015) and Camacho-Collados et al. (2016) exploited Wikipedia as a sense-annotated corpus using its hyperlinks. Other non-distributional approaches leveraged semantic networks to create representations solely based on the graph structure (Bordes et al., 2013; Wang et al., 2014b,a; Lin et al., 2015; Nickel et al., 2016). Finally, Rothe and Schütze (2015) and Pilehvar and Collier (2016) link existing word embeddings to lexical semantic resources to induce *sense embeddings* as a post-processing step.

None of these approaches, however, has created a joint semantic space of both lexical and semantic items starting from raw text while leveraging rich semantic resources. In this work we present SENSEMBED+, an approach which takes advantage equally from distributional information extracted from large amounts of raw data and existing knowledge from lexical semantic resources to jointly learn effective representations of words, word senses and synsets in the same semantic vector space.

Our contributions are threefold: (1) we propose a joint distributional and knowledge-based approach for obtaining low-dimensional continuous representations for word, word senses and synsets; (2) by leveraging the learned representations and lexical-semantic knowledge, we put forward a framework for many NLP tasks such as word, sense, and relational similarity and Word Sense Disambiguation with state-of-the-art performance on multiple datasets; (3) as a result, we also provide a wide comparison against alternatives to sense embedding representations.

6.1.1 SensEmbed+

In Chapter 3, we introduced SENSEMBED, a knowledge-based approach to create representations for individual word senses. By exploiting BabelNet (cf. Section 1.4), a multilingual encyclopedic dictionary and semantic network, SENSEMBED is capable of providing continuous vector representations of senses and offers an effective technique for measuring semantic similarity, showing the advantage gained by moving from the word to the sense level. SENSEMBED was created by disambiguating a large text corpus automatically by means of a knowledge-based WSD algorithm, and subsequently learning vector representations of the contexts of the words and senses in the corpus. Despite employing state-of-the-art disambiguation methods the WSD step came with an inherent limit: disambiguated words or collocations were replaced by their respective senses, losing the information of their occurrence. In Chapter 4 we introduced *Senses and Words to Vectors* (SW2V). a neural model which which speeded up the disambiguation step made by SENSEMBED while learning embeddings of both word a senses in the same vector space as an emerging feature. While SW2V did include both words and senses, their representations were of low quality compared to SENSEMBED. In addition, no synset embeddings were learned.

In order to alleviate these critical aspect of both previous approaches, we propose an extension which is able to produce a complete unified vector space model of meaning, with a consistent quality for words, senses and synsets. We also improve the supervised learning model of SENSEMBED by including more informative linguistic features from raw text. Our enhanced approach, that we call SENSEMBED+, is the main contribution presented in this paper, along with a long series of experiments to test the effectiveness of the SENSEMBED+ embeddings in solving a number of NLP tasks.

6.2 Model

We introduce SENSEMBED+, a method to automatically build a joint lexical and semantic space from a corpus of natural language text and a large-scale semantic network. The building method comprises two steps, namely collecting and disambiguating a text corpus, and computing the word, sense and synset vectors that represent the words in the corpus and their meaning.

6.2.1 Automatic Generation of a Sense-labeled Corpus

In order to learn reliable sense and synset embeddings we need a large corpus annotated with senses with a high degree of reliability. The assumption behind this step is that a corpus disambiguated with a tolerable amount of errors will offer enough information to learn high-quality semantic embeddings.

We apply a Word Sense Disambiguation algorithm (see Section 6.4 for details) to output, for each content word present in the input text, a sense from the inventory used for that word. We employ BabelNet as our sense inventory thanks to its richness in concepts and named entities across languages. For each sense prediction, we expect the disambiguation algorithm to also return a *disambiguation score*, i.e., a value in $[0, 1]$ indicating the degree of confidence that the algorithm had when disambiguating the target word. When dealing with multiword expressions such as "fire extinguisher" or named entity mentions such as "United Nations", we let the disambiguation algorithm disambiguate both the expression as a whole (if it has a recognized meaning), and its components. Consider for example the following sentence: *"The United Nations Children's Fund is a United Nations (UN) program that provides humanitarian assistance to children and mothers in developing countries."* The multiword expression *United Nations Children's Fund* is recognized as an entity (UNICEF), together with *United Nations* (the UN organization). Furthermore, *children* and *fund* are linked to the synsets representing their meaning. For the process of building our annotated corpus, in the case of disambiguated multiword expressions we consider only the longest sequences, discarding shorter fragments covered by them, again in line with the principle of building a high-precision resource.

Lexical and Semantic Sampling In the original version of SENSEMBED, after the WSD step, word occurrences which were properly disambiguated were replaced with their corresponding sense. This allows SENSEMBED to represent word senses at the expense of representing the original words. Only those word occurrences which were not disambiguated, due to uncertain context, were used as training instances for learning the corresponding word representation, therefore producing low-quality vectors. This unwelcome effect caused a loss of information that, instead, could be beneficial for a joint embedding model. To address this issue, SENSEMBED+ applies *lexical and semantic sampling*, thereby replacing multiple times each disambiguated word in the corpus with either the word itself, the chosen word sense or the corresponding synset ID with uniform probability. As a result, compared to the original SensEmbed, a richer input context is provided to the learning algorithm, with

Original	**Disambiguated**	**Sampling examples**	
In	In	In	In
mathematics	mathematics_bn:00053823n	mathematics_bn:00053823n	bn:00053823n
,	,	,	,
a	a	a	a
plane	plane_bn:00062766n	plane	plane_bn:00062766n
is	is	is	is
a	a	a	a
flat	flat_bn:00103058a	bn:00103058a	flat
two-dimensional	two-dimensional_bn:00108654a	bn:00108654a	two-dimensional
surface	surface_bn:00075373n	surface_bn:00075373n	surface

Table 6.1. From raw text to sense-annotated corpus. The `bn:<ID>` notation is used to encode the synset ID, whereas a sense of a word w is expressed as `w_bn:<ID>`.

a balanced distribution of words, senses and synsets across the corpus. The procedure is described in Algorithm 4, while examples of sampling steps are shown in Table 6.1 (in the two rightmost columns in the Table).

6.3 Learning the Semantic Vector Space Model

Once we have obtained a large corpus of text disambiguated with high precision, we employ it to learn a joint space of words, senses and synsets. We use the word2vec toolkit, an efficient neural network-based toolkit for learning high-quality vector representations of words that capture a large number of syntactic and semantic relations from the contexts in which the words occur. In the case of SENSEMBED+, we feed the neural network with the dataset produced as a result of the lexical and semantic sampling procedure explained in the previous section. For each target item (word, sense or synset label), a representation is computed by maximizing the log-likelihood of that item with respect to its context. We use the Continuous Bag of Words (CBOW) version of word2vec, where the training objective is to learn good representations that predict individual words given their context. Formally, given a sequence of items $s_1, s_2, \dots, s_T$, either words, senses or synsets, the objective of the model is to maximize the average log probability of the central item:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-W\leq i\leq W\\ i\neq 0}} logP\left(s_t|s_{t+i}\right) \tag{6.1}$$

where W is the size of the window defining the context to the left and right of the target

Algorithm 4 Lexical and Semantic Sampling algorithm

Input: sense-annotated corpus $\mathcal{C}$

Output: sense-annotated corpus with sampling $\mathcal{S}$

```
$\mathcal{S} \leftarrow \emptyset$
$i \leftarrow 4$
repeat
    for each $token \in \mathcal{C}$ do
        if $isSense(token)$ then
            rand ← a pseudo-random value between 0 and 1
            if rand < 0.33 then
                $\mathcal{S} \leftarrow \mathcal{S} \cup \{token\}$
            else if rand < 0.66 then
                $\mathcal{S} \leftarrow \mathcal{S} \cup \{wordOf(token)\}$
            else
                $\mathcal{S} \leftarrow \mathcal{S} \cup \{synsetOf(token)\}$
            end if
        else
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{token\}$
        end if
    end for
    $i \leftarrow i - 1$
until $i = 0$
return $\mathcal{S}$
```

word, typically in the range of five to ten words on each side. The result of the learning process is a set of vector-based semantic representations for each of the words, word senses and synsets in the sense-annotated corpus. An experimental justification of the use of CBOW compared to alternative tools for learning embeddings is provided in Section 6.4.3.

6.3.1 Measuring Word Similarity

The various techniques for computing vector representations are regularly tested against a series of evaluation benchmarks. Among these, tasks such as semantic relatedness and word similarity are consistently employed as means of evaluating word and sense embeddings. For word embeddings, the evaluation is usually straightforward: human judgments of relatedness between pairs of words are compared to a calculation given by a distance metric between the corresponding vectors, e.g., cosine similarity. This follows directly from the *distributional hypothesis*, which states that words related in meanings have similar representations in the distributional space (Rubenstein and Goodenough, 1965).

Algorithm 5 Expansion algorithm

Input: word w

Output: Set of related words, senses and synsets $\mathcal{S}_w$

```
$\mathcal{L} \leftarrow lemmasOf(w) \cup \{w\}$
$\mathcal{S}_w \leftarrow \mathcal{L}$
for each $l \in \mathcal{L}$ do
    for each $syn \in getSynsets(l)$ do
        for each $sen \in getSenses(syn)$ do
            $\mathcal{S}_w \leftarrow \mathcal{S}_w \cup \{sen\}$
            $\mathcal{S}_w \leftarrow \mathcal{S}_w \cup \{wordOf(sen)\}$
        end for
        $\mathcal{S}_w \leftarrow \mathcal{S}_w \cup \{syn\}$
    end for
end for
return $\mathcal{S}_w$
```

Associating senses with words In order to compute a similarity score between words that leverages the richer information contained in a joint word, sense and synset embedding model, we need an additional step to include also semantic vectors in the process. We call this step *expansion*, i.e., the process of associating a word with a set of vectors that represent words, senses or synsets. Expansion is done in four steps: (1) given a word, the vector of that word is obtained along with the vectors of all the senses of that word; (2) the word is lemmatized and the first step is repeated on the lemma; (3) we include all the synsets associated with either the word or the lemma in the BabelNet semantic network; (4) for each synset extracted in the previous step, all the word vectors of the English lexicalizations are considered as well. The full procedure is summarized in Algorithm 5.

For a given pair of words, we can now calculate a similarity score based on the vectors associated with them and with the additional vectors obtained with the expansion step. We follow the approach of Resnik (1999), who computes the similarity of words based on their senses by comparing the similarity of their closest senses among all possibilities. We extend this idea by including all the lemmas, inflected forms and synsets obtained by expanding the input words:

$$Sim\,(w_1, w_2) = \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} cos\,(\vec{s_1}, \vec{s_2}) \qquad (6.2)$$

where S_{w_i} is the set of words, senses and synsets associated to the word w_i resulting from Algorithm 5. The similarity score between vectors is then computed as their cosine similarity:

$$cos\,(\vec{s_1}, \vec{s_2}) = \frac{\vec{s_1} \cdot \vec{s_2}}{||\vec{s_1}||_2\ ||\vec{s_2}||_2} \tag{6.3}$$

Beyond distributional-based similarity A factor to consider when measuring semantic similarity with word embeddings computed from word co-occurrences is the possibility to get inaccurate results when low-frequency words are involved. Words that are observed only a few times are typically associated with less reliable vector representations, due to the lack of contextual information in the corpus. This in turn could affect the word similarity measure computed on the embeddings as in Formula 6.3. In order to mitigate this issue, we introduce semantic relatedness information about pairs of synsets from the BabelNet semantic network. Given a pair of synsets s_1 and s_2 and the set of synset relations provided by BabelNet $E = \{(s_i, s_j) : s_i \text{ is semantically related to } s_j\}$, we extend the similarity measure in Formula 6.2 to account for the presence of semantic relations between synsets associated with the input word pair as follows:

$$Sim^*(w_1, w_2) = \begin{cases} \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} cos\,(\vec{s_1}, \vec{s_2}) \times \beta, & \text{if } (s_1, s_2) \in E \\ \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} cos\,(\vec{s_1}, \vec{s_2}) \times \beta^{-1}, & \text{otherwise} \end{cases} \tag{6.4}$$

where $\beta \in [0, 1]$ is a parameter, which we call *Vicinity*, that controls the extent to which the new similarity measure leans on the BabelNet semantic network. By including the extra information from the semantic network, we are able to capture more accurate similarities between pairs of less frequent words. For example, the word pair *orthodontist-dentist* (taken from the SimLex-999 dataset of word pair similarity Hill et al. (2015)) is given a low score by the embedding-based similarity measure in Formula 6.2, despite common sense suggesting otherwise. However, the respective closest senses, `orthodontist_bn:00026279n` and `dentist_bn:00026279n` are connected in BabelNet by a hypernymy relation edge, therefore the updated similarity measure in Formula 6.4 results in a much higher score.

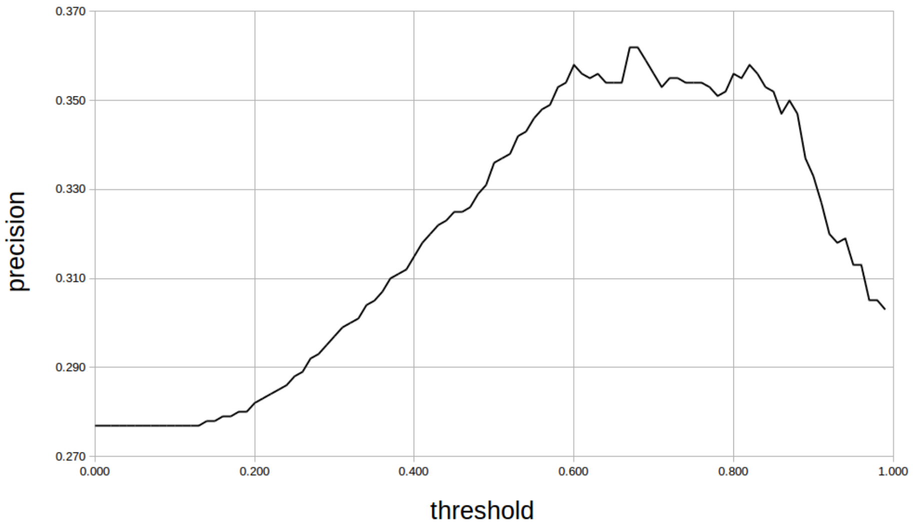

Figure 6.1. Precision of Babelfy on the Senseval-2 WSD Lexical Sample dataset when varying its confidence threshold over [0,1].

6.4 Experimental Setup

In this section, we describe the experimental setup of SENSEMBED+, including motivations for our choices, parameter tuning and comparison systems.

6.4.1 Disambiguation algorithm and tuning

To create a sense-annotated corpus, we utilized *Babelfy* (Moro et al., 2014), a state-of-the-art graph-based Word Sense Disambiguation (WSD) algorithm that performs WSD and entity linking jointly by leveraging the BabelNet semantic network. Babelfy works by modeling each concept in the semantic network according to its “semantic signature”, i.e., the set of the relevant vertices related to the concept, by applying a random walk algorithm on the graph. Given an input text in any of the languages supported by BabelNet, Babelfy constructs a semantic representation as a sub-graph of the semantic network using the signatures. Finally, an iterative process involving dense sub-graph heuristics computes the most likely sense of each word in the input text from its semantic representation. Note that each step in the disambiguation algorithm is inherently language independent, thus our approach can be easily ported to languages other than English.

We perform an additional step of fine-tuning the disambiguation algorithm by leveraging the confidence score returned by Babelfy for each disambiguated token. By using a reference corpus, we are able to test different values to cut off the less confident predictions and thus improving the overall accuracy of the disambiguation. More precisely, we test the precision of Babelfy on the Senseval 2 WSD Lexical Sample dataset (Edmonds and Cotton, 2001), a standard benchmark in WSD, varying the threshold value imposed on the confidence score to produce a sense label. As a result of this experiment, summarized in Figure 6.1, we observe a peak of precision in the range between 0.6 and 0.8, which justifies the use of a 0.7 threshold for the confidence score (which is, indeed, the default disambiguation threshold set in Babelfy). Since there is no statistical significance across the segment we kept the default value for the confidence score.

6.4.2 Corpus and Parameters

For building SENSEMBED+, we started from the September-2014 dump of the English Wikipedia, the same used in SENSEMBED, for comparison reasons. This edition of Wikipedia has approximately 4.6 million articles and contains roughly three billion words. As is shown in Algorithm 4, we performed the *Lexical and Semantic Sampling* four times on the resulting disambiguated corpus. By only including the sense labels predicted by Babelfy on the Wikipedia corpus that have a confidence score above the threshold, we obtained a sense-annotated dataset containing 12 million unique tokens with at least 5 occurrences, including 2.6 million word senses and 1.8 million synsets. We ran word2vec with the CBOW architecture, using a window size of 5 tokens and the remaining parameters as chosen in the original SENSEMBED: hierarchical softmax as our training algorithm, dimensionality set on 400 and the subsampling of frequent words set on 10^{-3}.

6.4.3 Choice of the embeddings training approach

In this Section, we motivate our choice of CBOW word2vec for building SENSEMBED+ by comparing it against alternative, commonly-used approaches to learning word embeddings.

- **word2vec**: the introduction of word2vec (Mikolov et al., 2013b) popularized word embeddings in Natural Language Processing. The tool, based on a simple feedforward neural network, is able to learn from large amounts of text corpora. Word2vec introduced two different models, namely Continuous Bag-of-Words (CBOW), and SkipGram. CBOW is based on a simple feedforward Neural Language Model, where

Model		RG65	MC30	WS353	WSSim	WSRel	SimLex	MEN	YP130	SimVerb	Avg
word2vec	CBOW	**0.829**	0.867	0.629	**0.724**	**0.487**	**0.471**	0.674	**0.703**	**0.468**	**0.650**
	SkipGram	0.810	**0.876**	0.613	0.708	0.446	0.466	0.677	0.723	0.465	0.643
fastText	[SkipGram]	0.771	0.686	**0.632**	0.660	0.551	0.489	**0.719**	0.701	0.415	0.625
	CBOW	0.654	0.636	0.441	0.466	0.393	0.458	0.541	0.600	0.432	0.513
GloVe		0.745	0.745	0.574	0.643	0.451	0.475	0.607	0.723	0.427	0.599
context2vec		0.531	0.696	0.526	0.632	0.347	0.395	0.570	0.614	0.356	0.519

Table 6.2. Performance of approaches for learning embeddings, in terms of Spearman correlation on nine standard word similarity datasets.

the projection layer averages all the context words. A log-linear classifier infers the word in the middle given its neighboring words. The second architecture. SkipGram is very similar to CBOW, but instead of predicting the word in the middle, it aims to infer the surrounding words given the central one.

- **GloVe**: the approach introduced by Pennington et al. (2014) provides a similar model to word2vec but, instead of using latent representations, it derives an explicit representation produced from statistical calculation directly from co-occurrence probabilities.
- **fastText**: introduced by Bojanowski et al. (2017), fastText can be seen as a natural extension of the SkipGram architecture of word2vec where character n-grams are learned and words are represented as the sum of the vectors of their constituents n-grams. The toolkit also includes an analogous extension for the CBOW architecture.
- **Context2vec**: finally, we consider the approach of Melamud et al. (2016), a model based on a biLSTM which learns both representations for word and sentences. It is based on the CBOW architecture of word2vec but instead of averaging the surrounding words for calculating the hidden state, two LSTMs, one on each side, are used to calculate the representation of the word context.

We evaluated the approaches using the same training corpus (cf. Section 6.4.2) on nine standard datasets for measuring word similarity and relatedness: RG65 introduced by Rubenstein and Goodenough (1965) made by word pairs of any POS; the WordSim-353 (Finkelstein et al., 2002, WS353), with its subsets proposed by Agirre and Soroa (2009) who divided WS353 in pairs that measure the degree of similarity (WSSim) and pairs that measure the degree of relatedness (WSRel); MC30 (Miller and Charles, 1991), a subset of 30 pairs from RG65 composed only by nouns; SimLex-999 (Hill et al., 2015), with 999 pairs, which puts special focus on representing antonyms as completely unrelated words; the MEN dataset introduced by Bruni et al. (2014) composed by 3000 pairs; two datasets especially made for analyzing verb similarity: YP130, created by Yang and Powers (2005), based on the WordNet taxonomy, and SimVerb-3500 (Gerz et al., 2016), a newer and much larger dataset extracted from the USF norms dataset[2] (Nelson et al., 2004) and VerbNet[3] (Kipper et al., 2008).

In Table 6.2 we present a comparison of the above embedding approaches on several datasets for word similarity, where Spearman correlation is calculated. fastText and CBOW show the best performance across datasets. Expectedly, default fastText, based on Skip-Gram, performs better than its CBOW alternative. An interesting result is the behavior of context2vec, in principle the most sophisticated model. Due to its configuration and the large amount of parameters learned at training time, it was impossible to learn embeddings with a large vocabulary, making this approach less competitive than its alternatives. Since the performance between CBOW and fastText is not significantly different across datasets, and on average CBOW is better, we chose CBOW for its simplicity and speed.

6.4.4 Comparison systems

In addition to the original SENSEMBED, we compare our enhanced SENSEMBED+ against five sense embedding models:

- **AutoExtend** (Rothe and Schütze, 2015): a system that combines word embeddings with semantic resources by learning embeddings of lexemes, senses and synsets from WordNet in a shared space. The embeddings are learned given the constraint that words are sums of their lexemes and synsets are sums of their lexemes. AutoExtend is based on an auto-encoder, a network that mimics the input vector at output layer.
- A **Unified Model for Word Sense Representation and Disambiguation** (Chen et al., 2014, UMWSRD) : a single model that uses WSD to inform the word sense

[2] http://w3.usf.edu/FreeAssociation/

[3] http://verbs.colorado.edu/verb-index

representations, and vice versa in a feedback loop. This approach exploits the structure of the sense inventory of WordNet.

- **Senses and Words to Vectors**[4] introduced in Chapter 4: A model which simultaneously learns embeddings for words and senses as an emerging feature, rather than via constraints, by exploiting knowledge from text corpora and BabelNet's semantic networks in a joint training phase.

- **NASARI** (Camacho-Collados et al., 2015b, 2016): a high-coverage multilingual vector representation which includes concepts and named entities from BabelNet. It exploits the multilingual semantic network and word2vec in order to learn representations of linguistic items in a unified space.

- **DeConf** (De-conflated Semantic Representations), introduced by Pilehvar and Collier (2016), is a technique which, based on an optimization function, decomposes a given set of pre-trained word representations into its constituent sense representations. The resulting sense embeddings appear near words closely related in the semantic network of WordNet.

6.5 Experimental Results

In this Section we report the experimental results obtained on a wide range of tasks such as word and relational similarity, word in context similarity, word-to-sense similarity, identification of linguistic properties, outlier detection and word sense disambiguation.

6.5.1 Word Similarity

Word similarity is the most widely used task for the evaluation of the representational power of word and sense embeddings. A dataset is given, composed of pairs of words associated with a numeric score produced by human judges indicating the extent to which the meaning of the two words is related. We test the embedding models listed in the previous section on the same datasets of word similarity in Section 6.4.3. The results, presented in Table 6.3,

[4] We used the 300-dimensional pre-trained word and sense embeddings from `http://lcl.uniroma1.it/sw2v`

Model	RG65	MC30	WS353	WSSim	WSRel	SimLex	MEN	YP130	SimVerb	Avg
AutoExtend	0.862	0.894	0.569	0.737	0.407	**0.562**	0.765	0.700	**0.529**	0.669
UMWSRD	0.869	**0.897**	0.545	0.691	0.389	0.465	0.650	0.724	0.472	0.634
SW2V	0.843	0.787	0.524	0.709	0.379	0.335	0.734	0.486	0.223	0.558
NASARI	0.799	0.726	0.502	0.623	0.317	0.234	0.541	0.280	0.154	0.464
DeConf	0.871	0.826	0.651	0.769	0.541	0.523	0.746	**0.738**	0.509	0.686
SensEmbed	0.865	0.851	0.682	0.731	0.555	0.404	0.763	0.573	0.389	0.646
SensEmbed+	**0.908**	0.866	**0.712**	**0.786**	**0.615**	0.513	**0.780**	0.728	0.516	**0.714**

Table 6.3. Performance of sense embeddings approaches, in terms of Spearman correlation on the task of word similiarity.

are given in terms of Spearman correlation between the relatedness scores predicted by the models and the human judgments, as done in most works.

SensEmbed+ obtains the best performance on this benchmark, significantly improving upon the previous version of the model, thanks to a richer and therefore more accurate encoding of the embeddings.

6.5.2 Word-to-Sense Similarity

While word pair similarity is a popular task, it is suboptimal for assessing the full potential of a joint model of lexical and semantic items. Here we consider the alternative task of word-to-sense similarity, which requires to output a relatedness score between a word and a sense from the WordNet inventory. Word and sense embedding models can be applied to this task in a straightforward fashion, since word vectors and sense vectors are represented in the same space, and thus they are directly comparable by means of a geometric distance metric. Jurgens et al. (2014, CLSS) introduced word-to-sense similarity as one of the tasks of their Cross-Level Semantic Similarity, an effort to test semantic similarity systems that are able to compare different types of text. The CLSS word-to-sense similarity dataset comprises 500 instances of words, each paired with a short list of candidate senses from WordNet with human ratings for their word-sense relatedness. Following the measures used for the task, the results are given as correlation scores (Pearson and Spearman), in Table 6.4. The performance of SensEmbed+ is higher than most systems, and on par with the other best performing model, AutoExtend.

Model	r	ρ
AutoExtend	**0.362**	0.364
UMWSRD	0.311	0.308
SW2V	0.130	0.146
NASARI	0.244	0.220
DeConf	0.211	0.250
SENSEMBED	0.316	0.333
SENSEMBED+	0.355	**0.369**

Table 6.4. Results of the evaluation on the word-to-sense similarity task in terms of Pearson (r) and Spearman (ρ) correlations.

6.5.3 Outlier Detection

The outlier detection task is a variation on the standard word similarity task. In this setting, a small set of words is given and the system must identify the one that does not belong to the group. Humans typically achieve almost perfect scores on this task while, interestingly, it is still a somewhat open problem for computers. For this experiment, we used the dataset provided with the 8-8-8 outlier detection task organized by Camacho-Collados and Navigli (2016), which comprises 8 groups of 8 words, each associated with 8 candidate outliers, ranked from the most to the least related to the original set. The evaluation script accepts vector representations for the input words and calculates the ranking of the candidate outliers based on their pairwise cosine similarity. We modified the script in order to accept multiple vectors for each word, according to the expansion procedure, and compute the word similarity considering the word senses as described in Section 6.3.1.

In the results on this task, presented in Table 6.5, we can see that SENSEMBED+ achieves a perfect score, in terms of both measures defined in the task (accuracy, and outlier position percentage, which considers the position of the outlier according to the proximity of the semantic cluster) matching the performance of the human annotators. This excellent result suggests that our joint word and sense embedding model is capable of representing more complex semantic relationships beyond pairwise relatedness.

6.5.4 Relational Similarity

Given two pairs of words $(w_{i,1}, w_{i,2})$ and $(w_{j,1}, w_{j,2})$, the degree of relatedness of the relation between $w_{i,1}$ and $w_{i,2}$ to the relation between $w_{j,1}$ and $w_{j,2}$ is called their *relational*

Model	**OPP score**	**Accuracy**
AutoExtend	82.8	37.5
UMWSRD	85.9	75.0
SW2V	48.4	37.5
NASARI	94.0	76.3
DeConf	93.8	62.5
SENSEMBED	98.0	95.3
SENSEMBED+	**100.0**	**100.0**

Table 6.5. Results of the evaluation on the outlier detection task (percentages).

similarity (Medin et al., 1990). The SemEval 2012 task on Measuring Degrees of Relational Similarity (Jurgens et al., 2012) provides a benchmark to evaluate systems that compute relational similarity, comprising 79 pairs of word relations, graded by human annotators. We test the word and sense embedding models against this gold standard dataset by computing a measure of relational similarity based on the cosine similarity of sense vectors Zhila et al. (2013):

$$Analogy(w_{i1}, w_{i2}, w_{j1}, w_{j2}) = \max_{\substack{s_{i1} \in \mathcal{S}_{w_{i1}} \\ s_{i2} \in \mathcal{S}_{w_{i2}} \\ s_{j1} \in \mathcal{S}_{w_{j1}} \\ s_{j2} \in \mathcal{S}_{w_{j2}}}} Sim(\vec{s_{i1}} - \vec{s_{i2}}, \vec{s_{j1}} - \vec{s_{j2}}) \tag{6.5}$$

where $\mathcal{S}_w$ is the set of senses associated with the word w. The results of the evaluation, shown in Table 6.6, are given in terms of MaxDiff score (Louviere, 1991) and Spearman correlation between the relational similarity scores predicted by the models and the human judgments. On this task, SenseEmbed+ outperforms all other competitors in terms of Spearman correlation, while attaining results on a par with DeConf when using the MaxDiff score.

6.5.5 Word in Context Similarity

While the word similarity benchmarks presented in the previous sections are widespread and useful tools to measure degrees of semantic relatedness, they are all hindered by the

Model	**MaxDiff**	ρ
AutoExtend	0.429	0.299
UMWSRD	0.462	0.379
SW2V	0.475	0.407
NASARI	0.443	0.306
DeConf	**0.506**	0.455
SensEmbed	0.471	0.394
SensEmbed+	**0.507**	**0.463**

Table 6.6. Results of the evaluation on the relational similarity task.

same issue, i.e., the words under consideration are isolated from their context, and thus they carry some degree of ambiguity that is not always straightforward to resolve. To overcome this issue, Huang et al. (2012) have proposed a dataset called Stanford Contextual Word Similarities, consisting of 2003 word pairs and their sentential contexts, with pairwise relatedness scores given by human judges. In this framework, we compute a similarity score for a pair of words in input based on the average similarity of their senses, with an additional weighting factor to take the contextual information into account:

$$avgSimC\left(w_1, w_2\right) = \frac{1}{|\mathcal{S}_{w_1}||\mathcal{S}_{w_2}|} \sum_{s_1 \in \mathcal{S}_{w_1}} \sum_{s_2 \in \mathcal{S}_{w_2}} Sim\left(\vec{s_1}, \vec{s_2}\right) \times P\left(s_1 | \mathcal{C}_{w_1}\right) \times P\left(s_2 | \mathcal{C}_{w_2}\right) \tag{6.6}$$

where $\mathcal{C}_{w_i}$ represents the context where the word w_i appears. For computing the vector we utilize a decay giving more weight to nearby words. Following Iacobacci et al. (2016) we use the exponential decay, based on the following formula:

$$\vec{C_{w_i}} = \sum_{\substack{j=I-W \\ j \neq I}}^{I+W} (1-\alpha)^{|I-j|-1} \vec{w_j} \tag{6.7}$$

where W is the context window taken into account, which we set to 30, I is the relative position of w_i in the sentence and $\alpha = 1 - 0.1^{(W-1)^{-1}}$ is the decay parameter. We choose

Model	MaxSim	AvgSim	AvgSimC
AutoExtend	**0.594**	0.645	0.653
UMWSRD	0.550	0.651	0.664
SW2V	0.519	0.626	0.448
NASARI	0.389	0.655	0.663
DeConf	0.579	0.655	0.664
SENSEMBED	0.567	0.639	0.688
SENSEMBED+	0.575	**0.659**	**0.703**

Table 6.7. Evaluation on the Stanford Contextual Word Similarity dataset in terms of Spearman.

the parameter in such a way that the immediate surrounding words contribute 10 times more than the last words on both sides of the window.

For comparison, we also compute two alternative word similarity measures that do not take context into account, as a means of evaluating the impact of the additional information on the task performance:

$$maxSim\,(w_1, w_2) = \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} Sim\,(\vec{s_1}, \vec{s_2}) \tag{6.8}$$

$$avgSim\,(w_1, w_2) = \frac{1}{|\mathcal{S}_{w_1}||\mathcal{S}_{w_2}|} \sum_{s_1 \in \mathcal{S}_{w_1}} \sum_{s_2 \in \mathcal{S}_{w_2}} Sim\,(\vec{s_1}, \vec{s_2}) \tag{6.9}$$

The results are given in Table 6.7. We comment on two main findings in this experiment. Despite being the main way to utilize sense embeddings for word similarity, the MaxSim (cf. Equation 6.8) similarity, introduced by Reisinger and Mooney (2010), performs lower than AvgSim (cf. Equation 6.9) similarity. Finally, as expected, AvgSimC (cf. Equation 6.6), which calculates a weighted average of all the sense vectors by taking into account the surrounding context, provides higher results than MaxSim and AvgSim in all configurations, with the exception of SW2V. Our approach reaches the best overall performance with the AvgSimC and ranks first in the AvgSim configuration.

6.5.6 Linguistic Properties

In order to gain a better insight into the nature of the embeddings introduced in this work, we performed an experiment aimed at measuring some of their intrinsic properties, as opposed to the task-based evaluations proposed in the previous sections. The QVEC evaluation framework (Tsvetkov et al., 2015) aims at measuring the quality of word embeddings by mapping the dimensions of the vector space to a manually constructed set of vectors whose dimensions represent specific linguistic properties based on WordNet. QVEC performs a one-to-many alignment of the components of the two vector spaces and computes an aggregated correlation score:

$$\text{QVEC} = \max_{A|\sum_j a_{ij} \leq 1} \sum_{i=1}^{D} \sum_{j=1}^{P} r(x_i, s_j) \times a_{ij} \tag{6.10}$$

where $A \in \{0,1\}^{D \times P}$ is the matrix of alignments, D is the word vector dimensionality, P is the amount of linguistic properties, $a_{ij} = 1$ iff x_i is aligned to s_j (0 otherwise), and $r(x_i, s_j)$ is the Pearson's correlation.

Model	**QVEC**
AutoExtend	0.511
UMWSRD	0.443
SW2V	0.463
NASARI	0.524
DeConf	0.511
SENSEMBED	0.504
SENSEMBED+	**0.537**

Table 6.8. QVEC score in the linguistic properties task.

We test the word and sense embedding models against QVEC and report the results in Table 6.8. While SENSEMBED performs quite well compared to the other models, SENSEMBED+ comes out as the embedding model that best aligns with linguistically interpretable dimensions.

6.5.7 Word Sense Disambiguation

Finally, we carried out experiments on Word Sense Disambiguation (WSD), a key task in lexical semantics. WSD is the task of assigning to each word in a given text a label indicating its sense (Navigli, 2009). We tested our models on two tasks, namely the prediction of the most frequent sense of a word, and the disambiguation of all the words in a text.

Most Frequent Sense Induction In the first experiment, we employ the word and sense embeddings to induce the most frequent sense (MFS) of the input words. This is a hard-to-beat baseline in WSD, typically computed by counting the word-sense pairs in an annotated corpus such as SemCor (Miller et al., 1993), which are limited in coverage, especially for languages other than English. In this experiment, we use the embeddings from a certain vector space model to determine the dominant sense of a target word as the closest sense vector in terms of cosine similarity to the vector of the target word, similarly to the computation of word-to-sense similarity in Section 6.5.2. The assumption here is that the more frequent a word sense, the more contextual information about that sense will be shared with the corresponding word embedding.

The test set was created by collecting all the words in SemCor with a minimum of five sense annotations (3731 words in total). We evaluated the MFS based on such annotations against induced dominant sense from the various sense embeddings approaches. We report the results in terms of precision @ K, i.e. when comparing the induced sense against the first K senses from WordNet (K = 1, 3, 5). The results, presented in Table 6.9, show how SENSEMBED+ is the best model to induce the dominant senses, although the overall performance of the models indicates that this is a hard problem.

All-Words Word Sense Disambiguation In a second experiment, we performed all-words WSD, i.e. a task whose goal is that of disambiguating all the content words in a given text. In order to learn models for disambiguating a large set of content words, a high-coverage sense-annotated corpus is required. Since all-words tasks do not usually provide any training data, the challenge here is not only to learn accurate disambiguation models from the training data, as is the case in the lexical sample task, but also to gather high-coverage training data and learn disambiguation models for as many words as possible.

Model	P@1	P@3	P@5
AutoExtend	22.8	52.0	56.6
UMWSRD	41.0	62.1	68.2
SW2V	39.7	60.3	67.5
NASARI	27.4	40.2	44.6
DeConf	30.1	55.8	64.3
SensEmbed	38.4	56.1	63.0
SensEmbed+	**42.7**	**63.4**	**70.2**

Table 6.9. Precision on the MFS task (percentages).

	Precision	F1
AutoExtend	0.667	0.726
UMWSRD	0.708	0.727
SW2V	0.625	0.486
NASARI	0.881	**0.818**
DeConf	0.592	0.637
SensEmbed	0.821	0.795
SensEmbed+	**0.902**	0.805

Table 6.10. Precision and F1 score in the SemEval 2007 Task 7: All-Words coarse-grained WSD.

We perform the WSD task with embeddings by 1) creating a context vector as a result of averaging the embedding vectors of the words surrounding the target word, 2) by choosing the most suitable sense for a target word which is most similar to the context vector:

$$sense(w, \mathcal{C}) = \arg\max_{s \in \mathcal{S}_w} Sim\left(\vec{s}, \vec{\mathcal{C}}\right) \tag{6.11}$$

where $\mathcal{S}_w$ is the set of senses associated with the target word w, and Sim is the similarity defined in Formula (6.2). We choose 0.2 as threshold following the NASARI parameter tuning (Camacho-Collados et al., 2016). We only output an answer when the confidence of the frameworks is larger than the threshold, without using any back-off strategy.

As our testing dataset we choose the SemEval-2007 English All-Words Coarse-Grained Word Sense Disambiguation Task (Navigli et al., 2007). The task consists of 2269 annotated words, 1108 nouns, 591 verbs, 362 adjectives and 208 adverbs. We report results in terms of Precision and and F1. We chose a coarse-grained dataset because a fine-grained WSD evaluation would not fit our purpose, due to our framework based on Formula 6.11 not being suitable for detecting subtle sense distinctions.

As shown in Table 6.10, SENSEMBED+ performs best together with the NASARI embedded vectors, with the former having higher precision and the latter showing slightly higher F1. All other sense-aware models deliver considerably lower performance on the task.

6.6 Analysis

We now analyse several aspects of SENSEMBED+ not covered in the experiment section.

Comparison against the Wikipedia 2018 corpus We first assess the impact that different versions of Wikipedia can have on SENSEMBED+, given the increase in size and richness over time. Therefore, in addition to the vectors introduced in the experimentation section, which were trained with the September 2014 dump of the English Wikipedia, we also trained a new set of vectors with the sense-annotated version of an up-to-date version of Wikipedia (May 2018) but with the same hyperparameter values mentioned in Section 6.4.

In Table 6.11 we show comparative statistics of the two corpora. Having up to 36% more unique tokens, the 2018 dump contains almost twice word senses and almost 50% more synsets. The sense coverage across parts of speech is considerably higher, especially in adverbs where coverage was increased by more than seven times.

Year	Tokens	Senses	Synsets	Nouns	Verbs	Adjectives	Adverbs	Total
2014	12331016	2642706	1838011	4357222	75765	41178	6552	7.6B
2018	16782882	4439069	2647036	6798698	141841	96697	48869	19.7B

Table 6.11. Statistics of the two sense-annotated corpora used to learn the SENSEMBED+ vectors, from September 2014 and May 2018 English Wikipedia dumps.

Year	WS353	Word2Sense		SCWS
	ρ	r	ρ	AvgSimC
2014	0.712	**0.355**	**0.369**	**0.703**
2018	**0.738**	0.333	0.336	0.657

Table 6.12. Comparison of performance on tasks with significant differences when Wikipedia 2014 and 2018 dumps are used for learning sense embeddings.

We performed experiments for comparing the performance between the new set of vectors and the version used in Section 6.4. We only found significant differences in a few experiments, which we report in Table 6.12. It seems the cost of having a larger lexical and semantic vocabulary causes a decrease in performance in some NLP tasks. This behavior could be due to the saturation of the vectors used to represented the embeddings. As was claimed by Erk and Padó (2008), a single vector "can only encode a fixed amount of structural information if its dimensionality is fixed".

The effect of Expansion and Vicinity In order to analyze the impact of the different components of our similarity measure introduced in Section 6.3.1, we carried out an experiment on the RG65 word similarity dataset (similar findings were obtained on all other word similarity datasets). In addition to the previously used models we included word embeddings vectors of GoogleNews[5], a set of word embeddings trained with word2vec, from a corpus of newspaper articles.

In Table 6.13 we show three different configurations for performing the word similarity task, namely Unique, Expansion and Expansion+Vicinity. The first configuration treats shared embeddings solely as word embeddings, by taking into account only the similarity between word vectors. Most of approaches reach highest performance by using this configuration. It is important to remark that both DeConf and AutoExtend start from word embeddings to induce sense embeddings in the same semantic space. We now move to the *Expansion* configuration. This configuration, when considered alone, seems harmful but it is needed for taking word senses into account for measuring similarity: this step enables the use of the Vicinity strategy, which requires the use of sense embeddings. The third

[5] `https://goo.gl/p4RXac`

Model	Rubenstein & Goodenough 65		
	Unique	Expansion	Expansion +Vicinity
Word2vec	0.674	0.673	0.673
AutoExtend	0.763	0.871	0.862
UMWSRD	0.746	0.819	0.869
SW2V	0.779	0.832	0.843
NASARI	0.710	0.702	0.799
DeConf	0.763	0.846	0.871
SENSEMBED	0.667	0.853	0.865
SENSEMBED+	**0.801**	**0.879**	**0.908**

Table 6.13. Spearman correlation on RG65 with different strategies.

configuration indeed shows better performance across all the models with the exception of AutoExtend, which justifies our use of Expansion+Vicinity in SENSEMBED+.

Closest Senses Finally, we performed a qualitative analysis of SENSEMBED+ vectors, aimed at assessing how well the semantic vector space separates the various meanings (senses and synsets) of ambiguous words. We show in Table 6.14 some of the closest words, senses and synsets to two ambiguous words: bar and $race$. For bar we include the two dominant senses: the meaning related to pub, defined by WordNet as *a room or establishment where alcoholic drinks are served over a counter* and the meaning of rod or $stick$, defined as *a rigid piece of metal or wood; usually used as a fastening or obstruction or weapon*. For the first sense, we see that the closest vector to the sense *bar* (bar_bn:00008462n) is the synset vector of bar (bn:00008462n). The next vectors in terms of proximity are bn:00014557n and cafe_bn:00014557n, which represent the synset and the sense of *coffee bar*, respectively, a concept deeply related with the concept of bar and, next, the sense and the synset of *eating place*, defined as *a building where people go to eat*. As we can see, SENSEMBED+ meaningfully represents lexical and semantic items in a shared space.

bar$_1^n$	***bar***$_3^n$	***race***$_3^n$	***race***$_3^n$	***race***$_1^n$
(pub)	(piece of metal)	(biology)	(acting)	(convertible car)
bar_bn:00008462n	bar_bn:00008464n	race_bn:00065800n	race_bn:00065798n	race_bn:00065799n
bn:00008462n	bn:00008464n	bn:00065800n	bn:00050342n	bn:00065799n
bn:00014557n	bn:00062446n	ethnicity	bn:00060837n	race
cafe_bn:00014557n	flange_bn:00035022n	gender_bn:00037634n	primaries_bn:00064361n	event
restaurant	bn:00011315n	bn:00031727n	bn:00066592n	bronze_medal_bn:00013330n
restaurant_bn:00029545n	hinge_bn:00035192n	ethnicity_bn:00031727n	programs_bn:00062899n	3,000_m_steeplechase_bn:00074144n
bn:00029545n	bn:00007536n	gender	doer_bn:00001177n	races_bn:00065799n
café_bn:00014557n	bn:00035192n	national_origin_bn:00019285n	u.s._senatorial_bn:00070459n	bn:00071626n
cafe	bn:00035022n	bn:00037634n	exit_polls_bn:00032244n	bn:00078199n
bar	axle	social_class	open_primary	marathon_race_bn:00053333n
coffee_shop	bn:00017487n	races_bn:00065800n	primary_election_bn:00064361n	grand_prix_bn:00041339n
lounge	pulley_bn:00011315n	religion_bn:00032770n	bn:00064361n	heats_bn:00043415n
coffee_shop_bn:00014557n	shaft	social_class_bn:00019478n	recounts_bn:00066592n	bronze_medal
bistro	coil	bn:00056964n	bn:00109704a	silver_medal
grill	bn:00023491n	religion	bn:00080384n	bn:00040935n
nightclub	bn:00062898n	race	united_state_senate_bn:00070459n	silver_medal_bn:00071626n
bn:00010725n	bn:00050864n	sexuality	bn:00103826a	sprint_event_bn:00025286n
nightclub_bn:00014419n	screw_bn:00069865n	ethnic_background_bn:00031722n	republican	gold_medal
bn:00014419n	dowel_bn:00028463n	skin_color_bn:00021378n	bn:00059193n	bn:00043419n
bn:00008206n	beam_bn:00009314n	sexuality_bn:00037634n	challenger	gold_medal_bn:00040935n

Table 6.14. The closest items (words, senses and synsets) to two ambiguous nouns: *bar* and *race*.

6.7 Conclusions

In this chapter we present SENSEMBED+, an approach which addresses the meaning conflation of word embedding representations and obtains continuous representations of both lexical items (i.e., words) and semantic items (i.e., word senses and synsets) in the same semantic space by leveraging distributional information from large amounts raw data and a vast multilingual encyclopedic dictionary and semantic network, i.e., BabelNet.

We evaluated our approach on several tasks including Word Similarity, Relational Similarity, Word Similarity In Context, Outlier Detection Linguistic Properties and Word Sense Disambiguation. SENSEMBED+ consistently achieves the best results or is in the same ballpark as the top-ranking approach. In addition we have shown the cohesion between our lexical and semantic embeddings by measuring word-in-context similarity, word-to-sense similarity, frequent sense induction and word sense disambiguation, all tasks which need explicit semantic information to be considered and therefore cannot be performed straightforwardly with vanilla word embeddings. We carried out evaluations against widely known sense representations and showing the advantages of our approach in various tasks with respect to state-of-the-art sense-based models.

Three conclusions can be drawn from the experimental results: (1) including words, senses and synsets in the joint space can significantly improve the effectiveness and accuracy of the resulting representations; (2) compared to the original SENSEMBED, by applying lexical and semantic sampling we obtain more accurate word embeddings in the same space as sense and synset embeddings; (3) SENSEMBED+ is not only an effective approach for similarity judgements but also particularly suitable for several other tasks which need the ability to discriminate word senses in effective way.

The SENSEMBED+ embeddings created with the Wikipedia 2014 and 2018 dumps are available at `http://lcl.uniroma1.it/sensembed_plus`.

Chapter 7

Joint BiLSTM-based Learning of Word and Sense Representations

> "whatever it is [an idea] that the mind can be employed about in thinking [..] is any immediate object of perception, thought, or understanding"
>
> *John Locke, 1689*

> "only in the context of a proposition [..] words have any meaning."
>
> *Gottlob Frege, 1884*

While word embeddings are now a de facto standard representation of words in most Natural Language Processing (NLP) tasks, recently the attention is shifting towards vector representations which capture the different meanings, i.e., senses, of words. While in the previous chapters all the models were based, or leveraged, feed-forward neural networks, in this chapter we explore the capabilities of a bidirectional Long Short Term Memory (LSTM) model to learn representations of word senses from order-aware contexts, while at the same time leveraging knowledge from existing semantic resources. We test our approach on various standard benchmarks for evaluating semantic representations, showing that our model achieves state-of-the-art performance across tasks, including the SemEval-2014 word-to-sense similarity task.

7.1 Introduction

Natural Language is inherently ambiguous, for reasons of communicative efficiency (Piantadosi et al., 2012). For us humans, ambiguity is not a problem, since we use common knowledge to understand each other. Therefore, a computational model suited to work side by side with humans without any supervision should deal with ambiguity to a certain extent. A necessary step to create such computer systems is to build formal representations of words and their meaning, either in the form of large repositories of knowledge, e.g., semantic networks, or as vectors in a geometric space.

In fact, representation learning has been a major research area in NLP in the past years, and latent vector-based representations, called *embeddings*, seem to be a good candidate to cope with ambiguity. Embeddings represent lexical and semantic items in a low-dimensional continuous space. These vector representations capture useful syntactic and semantic information of words and senses, such as regularities in the natural language, and relationships between them, in the form of relation-specific vector offsets. Recent approaches, such as word2vec (Mikolov et al., 2013a), and GloVe (Pennington et al., 2014), are capable of learning efficient word embeddings from large unannotated corpora. While word embeddings have paved the way to improvements in a countless number of NLP tasks Goldberg (2017), they still conflate the various meanings of each word and let its predominant sense prevail over the others in the resulting representation.

A strand of work aimed at tackling the lexical polysemy issue has proposed the creation of sense embeddings, i.e. embeddings which separate the various senses of each word in the vocabulary like in Huang et al. (2012); Chen et al. (2014) and in our work presented in Chapter 3, Chapter 4 and Chapter 6. One of the weaknesses of these approaches, however, is that they do not take word ordering into account during the learning process. On the other hand, word-based approaches based on RNNs that consider sequence information have

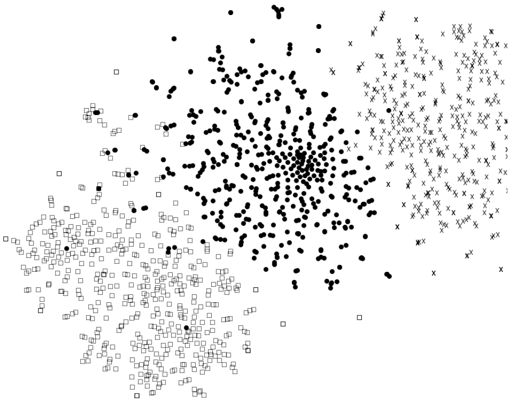

Figure 7.1. An example joint space where word vectors (squares) and sense vectors (dots and crosses) appear separated.

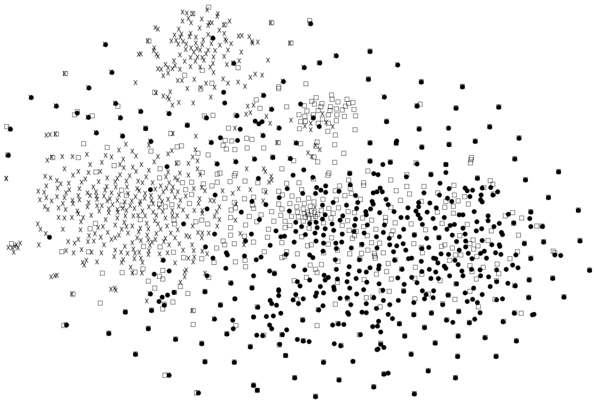

Figure 7.2. A shared space of words (squares) distributed across the space and two sense clusters (dots and crosses).

been presented but they are not competitive in terms of speed or quality of the embeddings (Mikolov et al., 2010; Mikolov and Zweig, 2012; Mesnil et al., 2013).

For example, in Figure 7.1 we show an excerpt of a tSNE projection of word and sense embeddings in the literature: as can be seen, first, the ambiguous word *bank* is located close to words which co-occur with it (squares in the Figure) and, second, the closest senses of bank (dots for the financial institution meaning and crosses for its geographical meaning) appear clustered in two separated regions without a clear correlation with (potentially ambiguous) words which are relevant to them. A more accurate representation would be to have word vectors distributed across all the space with defined clusters for each set of vectors related to each sense of a target word (Figure 7.2).

Recently, the much celebrated Long-Short Term Memory (LSTM) neural network model has emerged as a successful model to learn representations of sequences, thus providing an ideal solution for many Natural Language Processing tasks whose input is sequence-based, e.g., sentences and phrases (Hill et al., 2016; Melamud et al., 2016). However, to date LSTMs have not been applied to the effective creation of sense embeddings linked to an explicit inventory.

In this work, we address the open issues of current models that learn sense embeddings from sense-labeled corpora, by exploiting the capabilities of LSTMs, and present four main contributions:

- We introduce LSTMEmbed, an RNN model based on a bidirectional LSTM for learning word and sense embeddings in the same semantic space, which takes into account word ordering.

- We present an innovative idea for taking advantage of pre-existing embeddings using them as an objective during training.
- We show that LSTM-based models are suitable for learning not only contextual information, as is usually done, but also representations of individual words and senses.
- By linking our representations to a knowledge resource, we take advantage of the preexisting semantic information.

7.2 LSTMEmbed

Many approaches for learning embeddings are based on feed-forward neural networks (Section 2.2). However, recently LSTMs gained popularity in the NLP community as a new de facto standard model to represent natural language, by virtue of their context and word-order awareness. In this section we introduce LSTMEmbed, a novel method to learn word and sense embeddings jointly based on the LSTM architecture.

7.2.1 Model Overview

At the core of LSTMEmbed is a bidirectional Long Short Term Memory (BiLSTM), a kind of recurrent neural network (RNN), which uses a set of gates especially designed for handling long-range dependencies.

The bidirectional LSTM (BiLSTM) is a variant of the original LSTM (Hochreiter and Schmidhuber, 1997) particularly suited for temporal problems when access to the complete context is needed. In our case, we use an architecture similar to Kawakami and Dyer (2016), Kågebäck and Salomonsson (2016) and Melamud et al. (2016) where the state at each time step in the BiLSTM consists of the states of two LSTMs, centered in a particular timestep, accepting the input from previous timesteps in one LSTM, and the future timesteps in another LSTM. This is particularly suitable when the output corresponds to the analyzed timestep and not to the whole context.

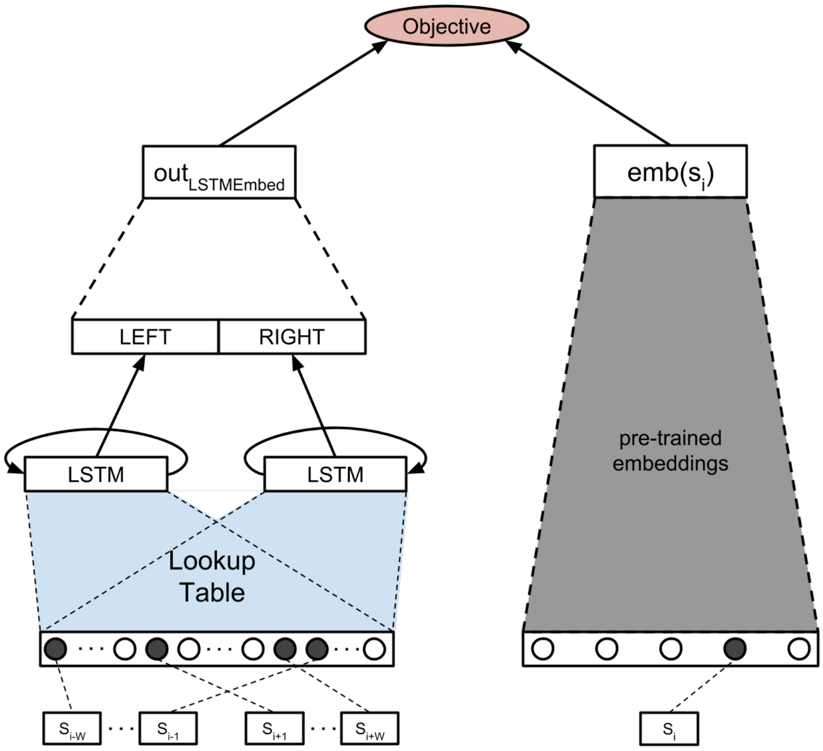

Figure 7.3. The LSTMEmbed architecture.

Figure 7.3 illustrates our model architecture. In marked contrast to the other LSTM-based approaches in the literature, we use sense-tagged text to provide input contexts of the kind $s_{i-W}, \ldots, s_i$ (the preceding context) and $s_i, \ldots, s_{i+W}$ (the posterior context), where s_j ($j \in [i-W, \ldots, i+W]$) is either a word or a sense tag from an existing inventory (see Section 7.3.1 for details). Each token is represented by its corresponding embedding vector $\mathbf{v}(s_j) \in \mathbb{R}^n$, given by a shared look-up table, which allows to learn representations taking into account the contextual information on both sides of the sentence. Next, the BiLSTM reads both sequences, i.e., the preceding context, from left to right, and the posterior context, from right to left:

$$\begin{aligned} o_l &= lstm_l(\mathbf{v}(s_{i-W}), ..., \mathbf{v}(s_{i-1})) \\ o_r &= lstm_r(\mathbf{v}(s_{i+1}), ..., \mathbf{v}(s_{i+W})) \end{aligned} \tag{7.1}$$

The model has one extra layer. The concatenation of the output of both LSTMs is projected linearly via a dense layer:

$$out_{LSTMEmbed} = \mathbf{W}^o(o_l \oplus o_r) \tag{7.2}$$

where $\mathbf{W}^o \in \mathbb{R}^{2m \times m}$ is the weights matrix of the dense layer with m being the dimension of the LSTM.

Then, the model compares $out_{LSTMEmbed}$ with $\mathbf{emb}(s_i)$, where $\mathbf{emb}(s_i)$ is a pretrained embedding vector of the target token (see Section 7.3.1 for an illustration of the pretrained embeddings that we use in our experiments). At training time, the weights of the network are modified in order to maximize the similarity between $out_{LSTMEmbed}$ and $\mathbf{emb}(s_i)$. The loss function is calculated in terms of cosine similarity:

$$loss = 1 - \mathcal{S}(\vec{v_1}, \vec{v_2}) = 1 - \frac{\vec{v_1} \cdot \vec{v_2}}{\|\vec{v_1}\| \|\vec{v_2}\|} \tag{7.3}$$

Once the training is over, we obtain latent semantic representations of words and senses jointly in the same vector space from the look-up table, i.e., the embedding matrix between the input and the LSTM, with the embedding vector of an item s given by $\mathbf{v}(s)$.

With respect to a standard BiLSTM, the novelties of LSTMEmbed can be summarized as follows:

- Using a sense-annotated corpus which includes both words and senses for learning the embeddings.
- A single look-up table, shared between both left and right LSTM, learned in the same architecture that represents both words and senses.
- A new learning method, which uses a set of pretrained embeddings as the objective for injecting semantic information.

7.3 Evaluation

We now present an experimental evaluation of the representations learned with LSTMEmbed. We first provide implementation details (Section 7.3.1), and then, to show the effectiveness of our model on a broad range of tasks, report on two sets of experiments: those involving sense-level tasks (Section 7.3.2) and those concerned with the word level (Section 7.3.3).

7.3.1 Implementation Details

Training data. We chose BabelNet (cf. Section 1.4) as our sense inventory. BabelNet is a large multilingual encyclopedic dictionary and semantic network, including approximately 16 million entries for concepts and named entities linked by semantic relations. As training corpus we used the English portion of BabelWiki[1], a multilingual corpus comprising the English Wikipedia. The corpus was automatically annotated with named entities and concepts using Babelfy (Moro et al., 2014), a state-of-the-art disambiguation and entity linking system, based on the BabelNet semantic network. The English section of BabelWiki contains 3 billion tokens and around 3 million unique tokens.

Learning embeddings. LSTMEmbed was built with the Keras[2] library using Theano[3] as backend. We trained our models with a Nvidia Titan X Pascal GPU. We set the dimensionality of the look-up table to 200 due to memory constraints. We discarded the 1,000 most frequent tokens and set the batch size to 2048. The training was performed for one epoch. As optimizer function we used Adaptive Moment Estimation or Adam[4].

As regards the objective embeddings $\mathbf{emb}(s_i)$ used for training (see Section 7.4.1), as a result of comparison against alternative approaches (reported in the analysis section below), we chose 400-dimension sense embeddings trained with word2vec's SkipGram architecture with negative sampling on the BabelWiki corpus with recommended parameters for the SkipGram architecture: window size of 10, negative sampling set on 10, sub-sampling of frequent words set to 10^{-3}.

7.3.2 Sense-based Evaluation

Our first set of experiments aims at showing the impact of our joint word and sense model in tasks where semantic, and not just lexical, relatedness is needed, namely Cross-Level Semantic Similarity and Most Frequent Sense Induction.

Comparison systems. We compare the performance of LSTMEmbed against alternative approaches to sense embeddings: SensEmbed (cf. Chapter 3), which obtained semantic

[1] `http://lcl.uniroma1.it/babelfied-wikipedia/`

[2] `https://keras.io`

[3] `https://goo.gl/RxRn5M`

[4] `https://goo.gl/fWWHBn`

Model	Pearson	Spearman
MeerkatMafia	0.389	0.380
SemantiKLU	0.314	0.327
SimCompass	0.356	0.344
AutoExtend	0.362	0.364
SensEmbed	0.316	0.333
Nasari	0.244	0.220
DeConf	0.349	0.356
LSTMEmbed	**0.380**	**0.400**

Table 7.1. Pearson and Spearman correlations on the CLSS word-to-sense similarity task.

representations by applying word2vec to the English Wikipedia disambiguated with Babelfy; Nasari (Camacho-Collados et al., 2015b, 2016), a technique for rich semantic representation of arbitrary concepts present in WordNet and Wikipedia pages; AutoExtend (Rothe and Schütze, 2015) which, starting from the word2vec word embeddings learned from GoogleNews[5], infers the representation of senses and synsets from WordNet; DeConf, an approach introduced by Pilehvar and Collier (2016) that decomposes a given word representation into its constituent sense representations by exploiting WordNet.

Experiment 1: Cross-Level Semantic Similarity. To best evaluate the ability of embeddings to discriminate between the various senses of a word, we opted for the SemEval-2014 task on Cross-Level Semantic Similarity (Jurgens et al., 2014, CLSS), which includes word-to-sense similarity as one of its sub-tasks. The CLSS word-to-sense similarity dataset comprises 500 instances of words, each paired with a short list of candidate senses from WordNet with human ratings for their word-sense relatedness. We include not only alternative sense-based representations but also the best performing approaches on this task: MeerkatMafia (Kashyap et al., 2014), which uses Latent Semantic Analysis (Deerwester et al., 1990) and WordNet glosses to get word-sense similarity measurements; SemantiKLU (Proisl et al., 2014), an approach based on a distributional semantic model trained on a large Web corpus from different sources; SimCompass (Banea et al., 2014), which combines word2vec with information from WordNet.

The results are given as Pearson and Spearman correlation scores in Table 7.1. LST-

[5] `https://goo.gl/p4RXac`

Model	P@1	P@3	P@5
AutoExtend	22.8	52.0	56.6
SensEmbed	38.4	56.1	63.0
Nasari	27.4	40.2	44.6
DeConf	30.1	55.8	64.3
LSTMEmbed	**39.0**	**59.2**	**66.0**

Table 7.2. Precision on the MFS task (percentages).

MEmbed achieves the state of the art by surpassing alternative sense embedding approaches as well as the best systems built specifically for the CLSS word-to-sense similarity task.

Experiment 2: Most Frequent Sense Induction. In a second experiment, we employed our representations to induce the most frequent sense (MFS) of the input words, which is known to be a hard-to-beat baseline for Word Sense Disambiguation (WSD) systems (Navigli, 2009). The MFS is typically computed by counting the word sense pairs in an annotated corpus such as SemCor (Miller et al., 1994).

To induce a MFS using sense embeddings, we identify – among all the sense embeddings of an ambiguous word – the sense which is closest to the word in terms of cosine similarity in the vector space. We evaluated all the sense embedding approaches on this task by comparing the induced most frequent senses against the MFS computed for all those words in SemCor which have a minimum number of 5 sense annotations (3731 words in total, that we release with the paper), so as to exclude words with insufficient gold-standard data for the estimates. Table 7.2 shows that LSTMEmbed fares better than alternative approaches in this task.

7.3.3 Word-based Evaluation

While our primary goal was to show the effectiveness of LSTMEmbed on tasks in need of sense information, we also carried out a second set of experiments focused on word-based evaluations with the objective of demonstrating the ability of our joint word and sense embedding model to tackle tasks traditionally approached with word-based models.

Model	**Accuracy**	
	TOEFL-80	ESL-50
Jauhar et al. (2015)	80.00	73.33
MSSG	78.26	57.14
Li and Jurafsky (2015)	82.61	50.00
MUSE	88.41	64.29
LSTMEmbed	**92.50**	**72.00**

Table 7.3. Synonym Recognition: accuracy (percentages).

Experiment 3: Synonym Recognition. We first experimented with synonym recognition: given a target word and a set of alternative words, the objective of this task is to select the member from the set which is most similar in meaning to the target word. The most likely synonym for a word w given the set of candidates $\mathcal{A}_w$ is calculated as:

$$Syn\left(w, \mathcal{A}_w\right) = \text{arg}\max_{v \in \mathcal{A}_w} Sim\left(w, v\right) \tag{7.4}$$

where Sim is the pairwise word similarity:

$$Sim\left(w_1, w_2\right) = \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} cosine\left(\vec{s_1}, \vec{s_2}\right) \tag{7.5}$$

where $\mathcal{S}_{w_i}$ is the set of words and senses associated with the word w_i. We consider all the inflected forms of every word, with and without all its possible senses.

In order to evaluate the performance of LSTMEmbed on this task, we carried out experiments on two datasets. The first one, introduced by Landauer and Dutnais (1997), is extracted directly from the synonym questions of the TOEFL (Test of English as a Foreign Language) questionnaire. The test is composed by 80 multiple-choice synonym questions with four choices per question. The second one, introduced by Turney (2001), provides a set of questions extracted from the synonym questions of the ESL test (English as a Second Language). Similarly to TOEFL, it is composed by 50 multiple-choice synonym questions with four choices per question.

Several related efforts used this kind of metric to evaluate their representations. We compare our approach with the following:

- Multi-Sense Skip-gram (Neelakantan et al., 2014, MGGS), an extension of the Skip-gram model of word2vec capable to learn multiple embeddings for a single word. The model makes no assumption about the number of prototypes.

- Li and Jurafsky (2015), a multi-sense embeddings model based on the Chinese Restaurant Process.

- Jauhar et al. (2015), a multi-sense approach based on expectation-maximization style algorithms for inferring word sense choices in the training corpus and learning sense embeddings while incorporating ontological sources of information.

- Modularizing Unsupervised Sense Embeddings (Lee and Chen, 2017, MUSE), an unsupervised approach that introduces a modularized framework to create sense-level representation learned with linear-time sense selection.

In Table 7.3 we report the performance of LSTMEmbed together with the alternative approaches (the latter obtained from the respective publications). We can see that LSTMEmbed outperforms all other approaches on this task.

Experiment 4: Outlier detection. Our second word-based evaluation is focused on outlier detection, a task intended to test the capability of the learned embeddings to create semantic clusters, that is, to test the assumption that the representation of related words should be closer than the representations of unrelated ones. We tested our model on the 8-8-8 dataset introduced by Camacho-Collados and Navigli (2016), containing eight clusters, each with eight words and eight possible outliers. In our case, we extended the similarity function used in the evaluation to consider both the words in the dataset and their senses, similarly to what we did in the synonym recognition task (cf. Equation 7.5). We can see in Table 7.4 that LSTMEmbed ranks second below SensEmbed in terms of both measures defined in the task (accuracy, and outlier position percentage, which considers the position of the outlier according to the proximity of the semantic cluster), with both approaches outperforming all other word-based and sense-based approaches.

Model	Corpus	Sense	8-8-8 OPP	8-8-8 Acc.
word2vec*	UMBC	-	92.6	73.4
	Wikipedia	-	93.8	70.3
	GoogleNews	-	94.7	70.3
GloVe*	UMBC	-	81.6	40.6
	Wikipedia	-	91.8	56.3
AutoExtend	GoogleNews	✓	82.8	37.5
SensEmbed	Wikipedia	✓	**98.0**	**95.3**
Nasari	Wikipedia	✓	94.0	76.3
DeConf	GoogleNews	✓	93.8	62.5
LSTMEmbed	Wikipedia	✓	96.1	78.1

Table 7.4. Outlier detection task (* reported in Camacho-Collados and Navigli (2016)).

7.4 Analysis

We now report on two kinds of analyses to, first, check if the use of pretrained embeddings is aiding to get better semantic representations and, second, to study the latent features from the embeddings learned by LSTMEmbed in terms of linguistic meaning.

7.4.1 Impact of Meaningful Objective Embeddings

The objective embedding $\mathbf{emb}$ we used in our work uses pretrained sense embeddings obtained from word2vec trained on BabelWiki, as explained in Section 7.3.1. Our assumption is that training with richer and meaningful objective embeddings enhances the representation delivered by our model in comparison to using word-based models. We put this hypothesis to test by comparing the performance of LSTMEmbed equipped with five sets of pretrained embeddings on a word similarity task. We used the WordSim-353 (Finkelstein et al., 2002, WS353) dataset, which comprises 353 word pairs annotated by human subjects with a pairwise relatedness score. We computed the performance of LSTMEmbed with the different pretrained embeddings in terms of Spearman correlation between the cosine similarities of the LSTMEmbed word vectors and the WordSim-353 scores.

The first set of pretrained embeddings is a 50-dimension word space model, trained with word2vec Skip-gram with the default configuration. The second set consists of the same

Model	Objective	Dim.	WS353
word2vec	-	-	0.488
GloVe	-	-	0.557
LSTMEmbed	random (baseline)	50	0.161
	word2vec	50	0.573
	word2vec + retro	50	0.569
	GoogleNews	300	0.574
	GloVe.6B	300	0.577
	SensEmbed	400	**0.612**

Table 7.5. Spearman correlation on the Word Similarity Task.

vectors, retrofitted with PPDB using the default configuration. The third is the GoogleNews set of pretrained embeddings. The fourth is the GloVe.6B word space model. Finally, we tested our model with the pretrained embeddings of SensEmbed. As baseline we include a set of normalized random vectors. As is shown in Table 7.5, using richer pretrained embeddings improves the resulting representations given by our model. All the representations obtain better results compared to word2vec and GloVe trained on the same corpus, with the sense embeddings from SensEmbed, a priori the richest set of pretrained embeddings, attaining the best performance.

7.4.2 Linguistic Properties

A second analysis is focused on assessing the quality of word embeddings by performing an intrinsic evaluation with the goal of discovering if the coordinates of the vectors representing a word have a linguistic meaning. To evaluate linguistic properties, we used the QVEC test introduced by Tsvetkov et al. (2015). QVEC uses a set of manually constructed word vectors whose components represent explicit linguistic properties. This task, shown to correlate strongly with performance in downstream tasks, evaluates each dimension of the vectors individually, instead of evaluating the quality of the model as a whole. In particular, QVEC maps each latent feature to at most one linguistic dimension.

The evaluation is based on component-wise correlations with the constructed set. QVEC aligns dimensions in a distributional word vector model with the linguistic dimension vectors to maximize the cumulative correlation of the aligned dimensions. The final score is the

Model	Dimensions				
	50	100	200	400	800
word2vec	0.257	0.309	0.361	0.427	0.532
GloVe	0.249	0.306	0.362	0.435	0.533
LSTMEmbed	**0.269**	**0.325**	**0.382**	**0.444**	**0.551**

Table 7.6. Progression of QVEC score over dimensionality for different word representations algorithms.

sum of the correlations of the aligned dimensions:

$$QVEC = \max_{A:\sum_j a_{ij} \leq 1} \sum_{i=1}^{D} \sum_{j=1}^{P} r(x_i, s_j) \times a_{ij} \tag{7.6}$$

where $A \in \{0,1\}^{D \times P}$ is the matrix of alignments, D the word vector dimensionality, P the amount of linguistic properties, $a_{ij} = 1$ iff x_i is aligned to s_j and $r(x_i, s_j)$ is the Pearson's correlation.

We performed this experiment over a range between 50 and 800 dimensions. QVEC is very dependent on dimensionality so we only compare same-dimension embeddings. Table 7.6 shows the progression of the QVEC score on embeddings of different dimensions trained with word2vec and GloVe, compared to LSTMEmbed. Overall, LSTMEmbed performs better than the word space models in this test. We observe that the gap between QVEC scores grows as the dimensionality increases, indicating a gain in the representational power of our model.

7.5 Conclusions

In this Chapter we present LSTMEmbed, a new model based on a bidirectional LSTM for learning embeddings of words and senses jointly which is able to learn semantic representations on par with or better than state-of-the-art approaches. We draw three main findings. Firstly, we have shown that our semantic representations are capable to represent properly the similarity between word and sense representations, showing state-of-the-art performance in the sense-aware tasks of word-to-sense similarity and most frequent sense induction. Secondly, our approach is also able to get good performance in standard word-based semantic evaluations, namely synonym recognition and outlier detection. Finally, the introduction of an output layer which predicts pretrained embeddings allows us to inject semantic information which improves the representations provided by LSTMEmbed.

We release the word and sense embeddings at the following URL: `http://cort.as/-A4wa`.

Chapter 8

Conclusions

In this thesis we highlight the importance of language in human society and how fundamental is its study. We emphasize the significance of natural language processing, which has, as its very first priority, to build efficient systems for analyzing and understanding human made text. We show that ambiguity is an inherent part of natural language enforcing the fact that autonomous systems capable to work together and interact with us should be able to handle ambiguity just as we do.

We highlight the importance of semantic representations which are fundamental for any NLP system. We introduce a set of models able to learn effective representation of lexical and semantic items. In addition, we show that those representations are perfectly suited to be used in many tasks and we have introduced a new way to take advantage of word embeddings to create better disambiguation tools.

In Chapter 3, we introduce SENSEMBED, a distributional model of semantic representations of word senses for effective word and relational similarity. We create a large and precise sense-annotated corpus by utilizing an out-of-the-box tool for Word Sense Disambiguation (WSD). We leverage the accurate corpus with a tool capable to learn embeddings and, by levering semantic knowledge from BabelNet, we create model capable to reach state-of-the-art performance on several semantic similarity tasks.

In Chapter 4, we introduce SW2V, a new model which by leveraging the semantic knowledge of BabelNet and large text corpora learns jointly word and sense embeddings in the same semantic space as an emerging feature, without constraining the representations.

In Chapter 5, we perform a deep analysis on several techniques to taking advantage of the semantic knowledge carried in word embeddings for improving Word Sense Disambiguation systems. We introduce four different strategies to combine standard features for WSD with embeddings trained with diverse strategies. We perform experiments on most commonly used benchmarks as Senseval-2, 3 and SemEval-2007 for Lexical Sample and All-words tasks, and in SemEval-2007 Lexical Substitution task reaching state-of-the-art performance.

Further, we present SENSEMBED+, a natural extension of our model presented in Chapter 3, which leverages a sense-annotated data in order to create effective representations of words, senses and synsets in the same semantic space. In Chapter 7, we go one step forward in semantic representation and demostrate the capabilities of Long Short Term Memory (LSTM) to learn effective representations of word senses. In contrast with the models introduced in Chapters 3, 4 and 6, this model, while training, takes into word ordering. In addition, this model put forward an innovative way to inject semantic knowledge by utilizing a set of pre-trained embeddings as objective of its training. Finally we show that model introduced in 6 and 7 built with the aid of WSD, are, at the same time, capable to perform the WSD task, closing the loop, showing that both, semantic representations and Word Sense Disambiguation are can mutually improve each other.

To summarize, our research provides a deeper insight to the vast field of distributional semantics by introducing several semantic representation approaches for many NLP tasks and, in particular, for the task of Word Sense Disambiguation. Our hope is that this work will contribute to the ultimate goal of NLP: a complete understanding of natural language.

Resources

Below we provide the set of resources developed during the PhD course.

Embeddings

SensEmbed Pretrained sense embeddings, trained with the 2014 dump of English Wikipedia, introduced in this thesis in Chapter 3. Formatted as plain text:

`http://lcl.uniroma1.it/sensembed/`

SW2V Pretrained word and sense embeddings and code used to build them, belonging to the work done in Chapter 4

`http://lcl.uniroma1.it/sw2v/`

SensEmbed+ Pretrained lexical and semantic embeddings, introduced in this thesis in Chapter 6. Formatted as plain text:

`http://lcl.uniroma1.it/sensembed_plus/`

Open source software

IMS Embed Java source code and runnable files from IMSEmbed, an extension of IMS (It Makes Sense) developed by Zhong and Ng (2010). The software corresponds to the work introduced in Chapter 5

`https://github.com/iiacobac/ims_wsd_emb`

LSTMEmbed Python source code used to build the models of LSTMEmbed. The software corresponds to the work introduced in Chapter 7

`https://bit.ly/2QhcwKO`

Bibliography

Eneko Agirre and Aitor Soroa. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece, 2009. URL http://www.aclweb.org/anthology/E09-1005.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, 2009. URL http://www.aclweb.org/anthology/N/N09/N09-1003.

Eneko Agirre, Aitor Soroa, and Mark Stevenson. Graph-based Word Sense Disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896, 2010. URL https://doi.org/10.1093/bioinformatics/btq555.

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. Random Walks for Knowledge-based Word Sense Disambiguation. *Computational Linguistics*, 40(1):57–84, 2014. URL http://dx.doi.org/10.1162/COLI_a_00164.

Jennifer E. Arnold, Thomas Wasow, Ash Asudeh, and Peter Alrenga. Avoiding attachment ambiguities: The role of constituent ordering. *Journal of Memory and Language*, 51(1): 55–70, 2004. URL http://www.stanford.edu/~wasow/AAWA.pdf.

Ben Athiwaratkun and Andrew Wilson. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 1645–1656, Vancouver, Canada, July 2017. URL `http://aclweb.org/anthology/P17-1151`.

Carmen Banea, Di Chen, Rada Mihalcea, Claire Cardie, and Janyce Wiebe. Simcompass: Using deep learning word embeddings to assess cross-level similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 560–565, Dublin, Ireland, 2014.

Marco Baroni and Roberto Zamparelli. Nouns are Vectors, Adjectives are Matrices: Representing Adjective-Noun Constructions in Semantic Space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, 2010. URL `http://www.aclweb.org/anthology/D10-1115`.

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 the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, 2014. URL `http://www.aclweb.org/anthology/P14-1023`.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003. URL `http://jmlr.org/papers/volume3/tmp/bengio03a.pdf`.

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 the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1524, Berlin, Germany, 2016. URL `http://www.aclweb.org/anthology/P16-1143`.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944937`.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational*

Linguistics, 5:135–146, 2017. URL https://transacl.org/ojs/index.php/tacl/article/view/999.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08)*, pages 1247–1250, Vancouver, Canada, 2008. URL https://doi.org/10.1145/1376616.1376746.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795, 2013. URL http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf.

Antoine Bordes, Sumit Chopra, and Jason Weston. Question Answering with Subgraph Embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar, October 2014. URL http://www.aclweb.org/anthology/D14-1067.

Sergey Brin and Michael Page. Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th Conference on World Wide Web (WWW7)*, pages 107–117, Brisbane, Australia, 1998. URL https://doi.org/10.1016/S0169-7552(98)00110-X.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research (JAIR)*, 49(1):1–47, 2014. URL http://dx.doi.org/10.1613/jair.4135.

Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, 2006. URL http://www.aclweb.org/anthology/J/J06/J06-1003.pdf.

Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. NUS-ML:Improving Word Sense Disambiguation Using Topic Features. In *Proceedings of the Fourth International Workshop on*

Semantic Evaluations (SemEval-2007), pages 249–252, Prague, Czech Republic, 2007. URL http://www.aclweb.org/anthology/S/S07/S07-1053.pdf.

José Camacho-Collados and Roberto Navigli. Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. In *Proceedings of the ACL Workshop on Evaluating Vector Space Representations for NLP*, pages 43–50, Berlin, Germany, 2016. URL http://www.aclweb.org/anthology/W/W16/W16-2508.pdf.

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 (Volume 1: Long Papers)*, volume 1, pages 741–751, Beijing, China, 2015a. URL http://www.aclweb.org/anthology/P15-1072.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. NASARI: a Novel Approach to a Semantically-Aware Representation of Items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 567–577, Denver, Colorado, May–June 2015b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1059.

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, 2016. URL http://lcl.uniroma1.it/nasari/papers/NASARI_AIJ.pdf.

Yee Seng Chan and Hwee Tou Ng. Word Sense Disambiguation with Distribution Estimation. In *Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI'05)*, pages 1010–1015, Edinburgh, Scotland, 2005. URL https://www.ijcai.org/Proceedings/05/Papers/1543.pdf.

Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. Improving distributed representation of word sense via wordnet gloss composition and context clustering. In *Proceedings of* *the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 15–20, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-2003`.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A Unified Model for Word Sense Representation and Disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar, 2014. URL `http://www.aclweb.org/anthology/D14-1110`.

Noam Chomsky. Syntactic structures. the hague: Mouton.. 1965. aspects of the theory of syntax. *Cambridge, Mass.: MIT Press.(1981) Lectures on Government and Binding, Dordrecht: Foris.(1982) Some Concepts and Consequences of the Theory of Government and Binding. LI Monographs*, 6:1–52, 1957.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT press, 1964.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 160–167, Helsinki, Finland, 2008. URL `https://dl.acm.org/citation.cfm?id=1390177`.

Bharath Dandala, Chris Hokamp, Rada Mihalcea, and Razvan Bunescu. Sense Clustering Using Wikipedia. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2013)*, pages 164–171, Hissar, Bulgaria, 2013. URL `http://www.aclweb.org/anthology/R13-1022`.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of American Society for Information Science*, 41(6):391–407, 1990. URL `https://doi.org/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9`.

Philip Edmonds and Scott Cotton. Senseval-2: Overview. In *The Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*, pages

1–5, Toulouse, France, 2001. URL http://dl.acm.org/citation.cfm?id=2387364.2387365.

Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D08-1094.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Measuring Word Meaning in Context. *Computational Linguistics*, 39(3):511–554, 2013. URL http://www.aclweb.org/anthology/J/J13/J13-3003.

Allyson Ettinger, Philip Resnik, and Marine Carpuat. Retrofitting Sense-Specific Word Vectors Using Parallel Text. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1378–1383, San Diego, California, 2016. URL http://www.aclweb.org/anthology/N16-1163.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, 2015. URL http://www.aclweb.org/anthology/N15-1184.

Ralph W Fasold and Jeff Connor-Linton. *An Introduction to Language and Linguistics*. Cambridge University Press, 2014.

Lev Finkelstein, Gabrilovich Evgeniy, Matias Yossi, Rivlin Ehud, Solan Zach, Wolfman Gadi, and Ruppin Eytan. Placing search in context: The concept revisited. *ACM Transactions on Information Systems (TOIS)*, 20(1):116–131, 2002. URL https://doi.org/10.1145/503104.503110.

J. R. Firth. A Synopsis of Linguistic Theory, 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32, 1957.

Lucie Flekova and Iryna Gurevych. Supersense Embeddings: A Unified Model for Supersense Interpretation, Prediction, and Utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041, Berlin, Germany, 2016. URL `http://www.aclweb.org/anthology/P16-1191`.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, 2013. URL `http://www.aclweb.org/anthology/N13-1092.pdf`.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas, 2016. URL `https://aclweb.org/anthology/D16-1235`.

Claudio Giuliano, Alfio Gliozzo, and Carlo Strapparava. FBK-irst: Lexical Substitution Task Exploiting Domain and Syntagmatic Coherence. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 145–148, Prague, Czech Republic, 2007. URL `http://www.aclweb.org/anthology/S/S07/S07-1029.pdf`.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. Random Walks and Neural Network Language Models on Knowledge Bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439, Denver, Colorado, 2015. URL `http://www.aclweb.org/anthology/N15-1165`.

Yoav Goldberg. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017. ISBN 1627052984, 9781627052986.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning Sense-specific Word Embeddings By Exploiting Bilingual Resources. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 497–507, Dublin, Ireland, 2014. URL http://aclweb.org/anthology/C14-1048.

Weiwei Guo and Mona 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*, pages 1542–1551, Uppsala, Sweden, 2010. URL http://www.aclweb.org/anthology/P10-1156.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 44–52, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S13-1005.

Zellig Harris. Distributional structure. *Word*, 10:146–162, 1954.

Samer Hassan and Rada Mihalcea. Semantic relatedness using salient semantic analysis. In *AAAI Conference on Artificial Intelligence*, 2011. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3616.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. UNT: SubFinder: Combining Knowledge Sources for Automatic Lexical Substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 410–413, Prague, Czech Republic, 2007. URL http://www.aclweb.org/anthology/S/S07/S07-1091.pdf.

Taher H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*, pages 517–526, Honolulu, Hawai, USA, 2002. URL https://doi.org/10.1145/511446.511513.

Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695, 2015. URL http://www.aclweb.org/anthology/J/J15/J15-4004.pdf.

Felix Hill, KyungHyun Cho, Anna Korhonen, and Yoshua Bengio. Learning to Understand Phrases by Embedding the Dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/711`.

Gerold Hintz and Chris Biemann. Language transfer learning for supervised lexical substitution. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 118–129. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1012. URL `http://aclweb.org/anthology/P16-1012`.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. URL `https://doi.org/10.1162%2Fneco.1997.9.8.1735`.

Charles F. Hockett and Charles D. Hockett. The origin of speech. *Scientific American*, 203(3):88–97, 1960. URL `http://psycnet.apa.org/doi/10.1038/scientificamerican0960-88`.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*, pages 545–554, Maui, Hawaii, USA, 2012. URL `https://doi.org/10.1145/2396761.2396832`.

Harry Hoijer. The Sapir-Whorf Hypothesis. In *Language in culture*, pages 92–105. Chicago: University of Chicago Press, 1954.

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. URL `https://doi.org/10.1016/j.artint.2012.10.002`.

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 50th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 873–882, Jeju Island, South Korea, 2012. URL http://www.aclweb.org/anthology/P12-1092.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. 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)*, volume 1, pages 95–105, Beijing, China, 2015. URL http://www.aclweb.org/anthology/P15-1010.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany, 2016. URL http://www.aclweb.org/anthology/P16-1085.

Matt Insall, Todd Rowland, and Eric W. Weisstein. “embedding", 2015. From MathWorld–A Wolfram Web Resource (access Sep 11, 2015) http://mathworld.wolfram.com/Embedding.html.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. Ontologically Grounded Multi-sense Representation Learning for Semantic Vector Space Models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 683–693, Denver, Colorado, 2015. URL http://www.aclweb.org/anthology/N15-1070.

Peng Jin, Diana McCarthy, Rob Koeling, and John Carroll. Estimating and Exploiting the Entropy of Sense Distributions. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 233–236, Boulder, Colorado, 2009. URL http://www.aclweb.org/anthology/N/N09/N09-2059.pdf.

Richard Johansson and Luis Nieto Piña. Embedding a Semantic Network in a Word Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 1428–1433, Denver, Colorado, 2015. URL http://www.aclweb.org/anthology/N15-1164.

David Jurgens, Peter D. Turney, Saif M. Mohammad, and Keith J. Holyoak. Semeval-2012 task 2: Measuring degrees of relational similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 356–364, 2012. URL http://www.aclweb.org/anthology/S12-1047.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. SemEval-2014 Task 3: Cross-Level Semantic Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, page 17, 2014. URL http://www.aclweb.org/anthology/S14-2003.

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, Osaka, Japan, 2016. URL http://aclweb.org/anthology/W16-5307.

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 (SemEval 2014)*, pages 416–423, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL http://www.aclweb.org/anthology/S14-2072.

Kazuya Kawakami and Chris Dyer. Learning to represent words in context with multilingual supervision. In *Proceedings of 4th International Conference of Learning Representations (Workshop Track)*, San Juan, Puerto Rico, 2016. URL http://arxiv.org/abs/1511.04623.

K. Kawano. *Warriors: Navajo Code Talkers*. Northland Publishing Company, 1990. ISBN 9780873585132. URL https://books.google.it/books?id=qLy6VOLdcVIC.

Boaz Keysar, Sayuri L Hayakawa, and Sun Gyu An. The Foreign-Language Effect: Thinking in a Foreign Tongue Reduces Decision Biases. *Psychological science*, 23(6):661–668, 2012. URL `https://doi.org/10.1177/0956797611432178`.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42 (1):21–40, 2008. URL `https://link.springer.com/article/10.1007/s10579-007-9048-2`.

Thomas K. Landauer and Susan T. Dutnais. 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. URL `http://www.stat.cmu.edu/~cshalizi/350/2008/readings/Landauer-Dumais.pdf`.

Guang-He Lee and Yun-Nung Chen. Muse: Modularizing unsupervised sense embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 327–337, Copenhagen, Denmark, September 2017. URL `https://www.aclweb.org/anthology/D17-1034`.

Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 41–48, 2002. URL `http://www.aclweb.org/anthology/W02-1006`.

Fritz Lehmann. Semantic Networks. *Computers & Mathematics with Applications*, 23(2–5):1–50, 1992. URL `http://www.sciencedirect.com/science/article/pii/089812219290135S`.

Omer Levy and Yoav Goldberg. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*, pages 171–180, Ann Arbor, Michigan, 2014. URL `http://www.aclweb.org/anthology/W14-1618`.

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, 2015. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570.

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*, pages 1722–1732, Lisbon, Portugal, 2015. URL http://aclweb.org/anthology/D15-1200.

Elizabeth D. Liddy. Natural language processing. In *Encyclopedia of Library and Information Science, 2nd Ed.* Marcel Decker, Inc, 2001. URL https://surface.syr.edu/istpub/63/.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2181–2187. AAAI Press, 2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2886521.2886624.

Quan Liu, Zhen-Hua Ling, Hui Jiang, and Yu Hu. Part-of-speech relevance weights for learning word embeddings. *CoRR*, abs/1603.07695, 2016. URL http://arxiv.org/abs/1603.07695.

Jordan Louviere. Best-Worst Scaling: A Model for the Largest Difference Judgments. Working paper, University of Alberta, 1991.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, Jun 1996. URL https://doi.org/10.3758/BF03204766.

Massimiliano Mancini, José Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. Embedding Words and Senses Together via Joint Knowledge-Enhanced Training. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 100–111, Vancouver, Canada, 2017. URL http://aclweb.org/anthology/K17-1012.

Chris Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. URL https://www.mitpressjournals.org/doi/10.1162/coli.2000.26.2.277.

Arthur B. Markman and Dedre Gentner. Structural Alignment during Similarity Comparisons. *Cognitive Psychology*, 25(4):431–467, 1993. URL http://dx.doi.org/10.1006/cogp.1993.1011.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7209-learned-in-translation-contextualized-word-vectors.pdf.

Diana McCarthy and Roberto Navigli. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic, 2007. URL http://www.aclweb.org/anthology/S/S07/S07-1009.pdf.

Douglas L. Medin, Robert L. Goldstone, and Dedre Gentner. Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, 1(1):64–69, 1990. URL https://doi.org/10.1111/j.1467-9280.1990.tb00069.x.

Oren Melamud, Ido Dagan, and Jacob Goldberger. Modeling Word Meaning in Context with Substitute Vectors. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 472–482, Denver, Colorado, May–June 2015a. URL http://www.aclweb.org/anthology/N15-1050.

Oren Melamud, Omer Levy, and Ido Dagan. A Simple Word Embedding Model for Lexical Substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado, June 2015b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W15-1501`.

Oren Melamud, Jacob Goldberger, and Ido Dagan. *context2vec*: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany, 2016. URL `http://www.aclweb.org/anthology/K/K16/K16-1006.pdf`.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. Investigation of Recurrent-neural-network Architectures and Learning Methods for Spoken Language Understanding. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH-2013)*, pages 3771–3775, Lyon, France, 2013. URL `http://www.isca-speech.org/archive/archive_papers/interspeech_2013/i13_3771.pdf`.

Rada Mihalcea and Andras Csomai. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM conference on Conference on Information and Knowledge Management (CIKM '07)*, pages 233–242, Lisbon, Portugal, 2007. URL `https://dl.acm.org/citation.cfm?id=1321475`.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. The Senseval-3 English lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain, 2004. URL `http://www.aclweb.org/anthology/W/W04/W04-0807.pdf`.

Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239, Miami, Florida, 2012. URL `http://ieeexplore.ieee.org/document/6424228/`.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH-*

2010), volume 2, page 3, Makuhari, Japan, 2010. URL `http://www.isca-speech.org/archive/archive_papers/interspeech_2010/i10_1045.pdf`.

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. *CoRR*, abs/1309.4168, 2013b. URL `http://arxiv.org/abs/1309.4168`.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, 2013c. URL `http://www.aclweb.org/anthology/N13-1090`.

George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995. URL `https://dl.acm.org/citation.cfm?id=219748`.

George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991. URL `https://doi.org/10.1080/01690969108406936`.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. A Semantic Concordance. In *Proceeding of ARPA Human Language Technology Workshop*, pages 303–308, Plainsboro, New Jersey, 1993. URL `http://www.aclweb.org/anthology/H/H93/H93-1061.pdf`.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. Using a Semantic Concordance for Sense Identification. In *Proceeding of ARPA Human Language Technology Workshop*, pages 240–243, Plainsboro, New Jersey, 1994. URL `http://www.aclweb.org/anthology/H/H94/H94-1046.pdf`.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of COLING 2012*, pages 1781–1796, Mumbai, India, December 2012. URL http://www.aclweb.org/anthology/C12-1109.

Jeff Mitchell and Mark Steedman. Orthogonality of syntax and semantics within distributional spaces. 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 1301–1310, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1126.

Andriy Mnih and Geoffrey Hinton. Three New Graphical Models for Statistical Language Modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 641–648, Corvalis, Oregon, USA, 2007. URL http://doi.acm.org/10.1145/1273496.1273577.

Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3583-a-scalable-hierarchical-distributed-language-model.pdf.

Saif Mohammad and Graeme Hirst. Determining Word Sense Dominance Using a Thesaurus. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL '06)*, pages 121–128, Trento, Italy, 2006. URL http://www.aclweb.org/anthology/E/E06/E06-1016.pdf.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistic*, 2:231–244, 2014. URL http://www.aclweb.org/anthology/Q14-1019.

Charles Morris. The measurement of meaning. charles e. osgood , george j. suci , percy h. tannenbaum. *American Journal of Sociology*, 63(5):550–551, 1958.

Nikola Mrksic, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324, 2017. URL https://transacl.org/ojs/index.php/tacl/article/view/1171.

Eric Nalisnick and Sachin Ravi. Infinite dimensional word embeddings. In *Proceedings of 5th International Conference on Learning Representations (Workshop Track)*, Toulon, France, 2017. URL https://openreview.net/pdf?id=SyGmHLfte.

Roberto Navigli. Word Sense Disambiguation: A Survey. *ACM Computing Surveys (CSUR)*, 41(2):1–69, 2009. URL http://doi.acm.org/10.1145/1459352.1459355.

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. URL https://dl.acm.org/citation.cfm?id=2397579.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, 2007. URL http://www.aclweb.org/anthology/S/S07/S07-1006.

Roberto Navigli, David Jurgens, and Daniele Vannella. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA, 2013. URL http://www.aclweb.org/anthology/S13-2040.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Process-*

ing (EMNLP), pages 1059–1069, Doha, Qatar, 2014. URL http://aclweb.org/anthology/D14-1113.

Douglas L. Nelson, Cathy L. McEvoy, and Thomas A. Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.

Hwee Tou Ng and Hian Beng Lee. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, Santa Cruz, California, 1996. URL http://www.aclweb.org/anthology/P96-1006.

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 the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany, 2016. URL http://anthology.aclweb.org/P16-2074.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1955–1961. AAAI Press, 2016. URL http://dl.acm.org/citation.cfm?id=3016100.3016172.

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. URL https://doi.org/10.1017/S135132490500402X.

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. URL http://www.aclweb.org/anthology/E17-1009.

Siddharth Patwardhan and Ted Pedersen. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006*

Workshop Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together, volume 1501, pages 1–8, Trento, Italy, 2006. URL http://www.aclweb.org/anthology/W/W06/W06-2501.pdf.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. URL http://www.aclweb.org/anthology/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. URL http://aclweb.org/anthology/N18-1202.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 21–26, Beijing, China, 2015. URL http://www.aclweb.org/anthology/P15-2004.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. The communicative function of ambiguity in language. *Cognition*, 122(3):280 – 291, 2012. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2011.10.004. URL http://www.sciencedirect.com/science/article/pii/S0010027711002496.

Mohammad Taher Pilehvar and Nigel Collier. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1174.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 1341–1351, Sofia, Bulgaria, 2013. URL http://www.aclweb.org/anthology/P13-1132.

Mohammad Taher Pilehvar, José Camacho-Collados, Roberto Navigli, and Nigel Collier. Towards a Seamless Integration of Word Senses into Downstream NLP Applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1857–1869, Vancouver, Canada, 2017. URL http://aclweb.org/anthology/P17-1170.

Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th ACL*, pages 1522–1531, Uppsala, Sweden, 2010. URL http://dl.acm.org/citation.cfm?id=1858681.1858835.

Barry B. Powell. *What is Writing?*, pages 11–18. Wiley-Blackwell, 2012. URL http://dx.doi.org/10.1002/9781118293515.ch1.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. SemEval-2007 Task-17: English Lexical Sample, SRL and All Words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic, 2007. URL http://www.aclweb.org/anthology/S/S07/S07-1016.pdf.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2025.

Thomas Proisl, Stefan Evert, Paul Greiner, and Besim Kabashi. Semantiklue: Robust semantic similarity at multiple levels using maximum weight matching. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 532–540, Dublin, Ireland, 2014. URL http://www.aclweb.org/anthology/S14-2093.

Lin Qiu, Yong Cao, Zaiqing Nie, Yong Yu, and Yong Rui. Learning word representation considering proximity and ambiguity. In *Proceedings of the Twenty-Eighth AAAI Con-*

ference on Artificial Intelligence, AAAI'14, pages 1572–1578. AAAI Press, 2014. URL `http://dl.acm.org/citation.cfm?id=2892753.2892771`.

Lin Qiu, Kewei Tu, and Yong Yu. Context-Dependent Sense Embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 183–191, Austin, Texas, 2016. URL `https://aclweb.org/anthology/D16-1018`.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain, 2017. URL `http://www.aclweb.org/anthology/E17-1010`.

Joseph Reisinger and Raymond J. Mooney. Multi-Prototype Vector-Space Models of Word Meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California, 2010. URL `http://www.aclweb.org/anthology/N10-1013`.

Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, (IJCAI '95)*, volume 1, pages 448–453, 1995. URL `http://ijcai.org/Proceedings/95-1/Papers/059.pdf`.

Philip Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11(1):95–130, 1999. URL `https://dl.acm.org/citation.cfm?id=3013547`.

Bryan Rink and Sanda Harabagiu. UTD: Determining relational similarity using lexical patterns. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 413–418, Montreal, Canada, 2012. URL `http://aclweb.org/anthology/S12-1055`.

Bryan Rink and Sanda Harabagiu. The Impact of Selectional Preference Agreement on Semantic Relational Similarity. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS) – Long Papers*, pages 204–215, Potsdam, Germany, 2013. URL http://www.aclweb.org/anthology/W13-0118.

Douglas Roland, Jeffrey L. Elman, and Victor S. Ferreira. Why is that? Structural prediction and ambiguity resolution in a very large corpus of English sentences. *Cognition*, 98(3):245–272, 2006. URL https://doi.org/10.1016/j.cognition.2004.11.008.

Sascha Rothe and Hinrich Schütze. AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. 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 1793–1803, Beijing, China, 2015. URL http://www.aclweb.org/anthology/P15-1173.

Herbert Rubenstein and John B. Goodenough. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633, 1965. URL https://dl.acm.org/citation.cfm?id=365657.

Hinrich Schütze. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1): 97–124, 1998. URL https://dl.acm.org/citation.cfm?id=972724.

Roy Schwartz, Roi Reichart, and Ari Rappoport. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning (CoNLL 2015)*, pages 258–267, Beijing, China, 2015. URL http://www.aclweb.org/anthology/K15-1026.

Benjamin Snyder and Martha Palmer. The english all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain, 2004. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W04/W04-0811.pdf.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1170.

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. URL http://www.aclweb.org/anthology/S17-2008.

Jiri Stetina, Sadao Kurohashi, and Makoto Nagao. General Word Sense Disambiguation Method Based on a Full Sentential Context. In *Usage of WordNet in Natural Language Processing, Proceedings of COLING-ACL Workshop*, Montreal, Quebec, Canada, 1998. URL http://www.aclweb.org/anthology/W/W98/W98-0701.pdf.

György Szarvas, Chris Biemann, and Iryna Gurevych. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1131–1141, Atlanta, Georgia, 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N13-1133.

Kaveh Taghipour and Hwee Tou Ng. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing, China, July 2015a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K15-1037.

Kaveh Taghipour and Hwee Tou Ng. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. In *Proceedings of the 2015 Annual Conference of the NAACL*, pages 314–323, Denver, Colorado, 2015b. URL http://www.aclweb.org/anthology/N15-1035.

Thesaurus.com. Roget's New Millennium™, 2007. Thesaurus, First Edition (v 1.3.1). Lexico Publishing Group, LLC http://thesaurus.reference.com.

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 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160, Dublin, Ireland, August 2014. URL `http://www.aclweb.org/anthology/C14-1016`.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of Word Vector Representations by Subspace Alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1243`.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858681.1858721`.

Peter D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (EMCL '01)*, pages 491–502, Freiburg, Germany, 2001. URL `https://dl.acm.org/citation.cfm?id=650004`.

A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

Tim Van de Cruys and Marianna Apidianaki. Latent Semantic Word Sense Induction and Disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1476–1485, Portland, Oregon, USA, 2011. URL `http://www.aclweb.org/anthology/P11-1148`.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. Latent Vector Weighting for Word Meaning in Context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1012–1022, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2145432.2145542`.

Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *CoRR*, abs/1412.6623, 2014. URL `http://arxiv.org/abs/1412.6623`.

Simon Šuster, Ivan Titov, and Gertjan van Noord. Bilingual Learning of Multi-sense Embeddings with Discrete Autoencoders. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1346–1356, San Diego, California, 2016. URL `http://www.aclweb.org/anthology/N16-1160`.

Thuy Vu and D. Stott Parker. K-Embeddings: Learning Conceptual Embeddings for Words using Context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1262–1267, San Diego, California, June 2016. URL `http://www.aclweb.org/anthology/N16-1151`.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1112–1119. AAAI Press, 2014a. URL `http://dl.acm.org/citation.cfm?id=2893873.2894046`.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge Graph and Text Jointly Embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, October 2014b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1167`.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. Structured Training for Neural Network Transition-Based Parsing. 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 323–333, Beijing, China, July 2015. URL `http://www.aclweb.org/anthology/P15-1032`.

Zhaohui Wu and C. Giles. Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia. In *AAAI Conference on Artificial Intelligence*, San

Francisco, California, 2015. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9878.

Dongqiang Yang and David M. W. Powers. Measuring Semantic Similarity in the Taxonomy of WordNet. In *Proceedings of the Twenty-eighth Australasian Conference on Computer Science - Volume 38*, pages 315–322, Newcastle, Australia, 2005. URL http://dl.acm.org/citation.cfm?id=1082161.1082196.

Mo Yu and Mark Dredze. Improving Lexical Embeddings with Semantic Knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550, Baltimore, Maryland, 2014. URL http://www.aclweb.org/anthology/P14-2089.

Deniz Yuret. Ku: Word sense disambiguation by substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 207–213, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1621474.1621518.

Torsten Zesch, Christof Müller, and Iryna Gurevych. Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 2, pages 861–866, Chicago, Illinois, 2008. ISBN 978-1-57735-368-3. URL http://dl.acm.org/citation.cfm?id=1620163.1620206.

Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. Combining Heterogeneous Models for Measuring Relational Similarity. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1000–1009, Atlanta, Georgia, 2013.

Zhi Zhong and Hwee Tou Ng. It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-4014.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Confer-*

ence on Empirical Methods in Natural Language Processing, pages 1393–1398, Seattle, Washington, USA, October 2013. URL `http://www.aclweb.org/anthology/D13-1141`.