SAPIENZA
UNIVERSITÀ DI ROMA

New Frontiers in Supervised Word Sense Disambiguation: Building Multilingual Resources and Neural Models on a large scale

Department of Computer Science

Dottorato di Ricerca in Informatica – XXX Ciclo

Candidate

Alessandro Raganato
ID number 1254144

Thesis Advisor

Prof. Roberto Navigli

Thesis defended on 12 February 2018
in front of a Board of Examiners composed by:

Prof. Nicola Leone (Università della Calabria)
Prof. Gianluca Foresti (Università di Udine)
Prof. Sara Foresti (Università di Milano)

The thesis has been peer reviewed by:

Prof. Anders Søgaard (University of Copenhagen)
Prof. Chris Biemann (TU Darmstadt)

New Frontiers in Supervised Word Sense Disambiguation: Building Multilingual Resources and Neural Models on a large scale
Ph.D. thesis. Sapienza – University of Rome

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

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

Abstract

Word Sense Disambiguation is a long-standing task in Natural Language Processing (NLP), lying at the core of human language understanding. While it has already been studied from many different angles over the years, ranging from knowledge based systems to semi-supervised and fully supervised models, the field seems to be slowing down in respect to other NLP tasks, e.g., part-of-speech tagging and dependencies parsing. Despite the organization of several international competitions aimed at evaluating Word Sense Disambiguation systems, the evaluation of automatic systems has been problematic mainly due to the lack of a reliable evaluation framework aiming at performing a direct quantitative confrontation.

To this end we develop a unified evaluation framework and analyze the performance of various Word Sense Disambiguation systems in a fair setup. The results show that supervised systems clearly outperform knowledge-based models. Among the supervised systems, a linear classifier trained on conventional local features still proves to be a hard baseline to beat. Nonetheless, recent approaches exploiting neural networks on unlabeled corpora achieve promising results, surpassing this hard baseline in most test sets. Even though supervised systems tend to perform best in terms of accuracy, they often lose ground to more flexible knowledge-based solutions, which do not require training for every disambiguation target. To bridge this gap we adopt a different perspective and rely on sequence learning to frame the disambiguation problem: we propose and study in depth a series of end-to-end neural architectures directly tailored to the task, from bidirectional Long Short-Term Memory to encoder-decoder models. Our extensive evaluation over standard benchmarks and in multiple languages shows that sequence learning enables more versatile all-words models that consistently lead to state-of-the-art results, even against models trained with engineered features.

However, supervised systems need annotated training corpora and the few available to date are of limited size: this is mainly due to the expensive and time-consuming process of annotating a wide variety of word senses at a reasonably high scale, i.e., the so-called knowledge acquisition bottleneck. To address this issue, we also present different strategies to acquire automatically high quality sense annotated data in multiple languages, without any manual effort. We assess the quality of the sense annotations both intrinsically and extrinsically achieving competitive results on multiple tasks.

Publications

2017

- **Alessandro Raganato**, Claudio Delli Bovi and Roberto Navigli. *Neural Sequence Learning Models for Word Sense Disambiguation.* Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1167–1178, 7-11 September 2017.

- Simone Papandrea, **Alessandro Raganato** and Claudio Delli Bovi. *SupWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation.* Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations, pages 103–108, 7-11 September 2017.

- Claudio Delli Bovi, José Camacho Collados, **Alessandro Raganato** and Roberto Navigli. *EuroSense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text.* Proceedings of the 55th annual meeting of the Association for Computational Linguistics (ACL), pages 594–600, 30 July-4 August 2017.

- Claudio Delli Bovi and **Alessandro Raganato**. *Sew-Embed at SemEval-2017 Task 2: Language-Independent Concept Representations from a Semantically Enriched Wikipedia.* Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 252–257, 30 July-4 August 2017.

- **Alessandro Raganato**, José Camacho Collados and Roberto Navigli. *Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison.* Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL) 2017, pages 99–110, Valencia, Spain, 3-7 April 2017.

2016

- **Alessandro Raganato**, José Camacho Collados, Antonio Raganato and Yunseo Joung. *Semantic Indexing of Multilingual Corpora and its Application on the History Domain.* Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), pages 140–147, COLING 2016, Osaka, Japan.

- **Alessandro Raganato**, Claudio Delli Bovi and Roberto Navigli. *Automatic Construction and Evaluation of a Large Semantically Enriched Wikipedia.* Proceedings of 25th International Joint Conference on Artificial Intelligence (IJCAI), pages 2894–2900, 9-15 July 2016.

- José Camacho Collados, Claudio Delli Bovi, **Alessandro Raganato** and Roberto Navigli. *A Large-Scale Multilingual Disambiguation of Glosses.* Proceedings of the

10th International Conference on Language Resources and Evaluation (LREC), pages 1701–1708, 23-28 May 2016.

2015

- Federico Scozzafava, **Alessandro Raganato**, Andrea Moro and Roberto Navigli. *Automatic Identification and Disambiguation of Concepts and Named Entities in the Multilingual Wikipedia*. Proceedings of the 14th Congress of the Italian Association for Artificial Intelligence (AI*IA 2015), pages 357–366, Ferrara, Italy, September 23-25th, 2015.

Contents

Chapter 1

Introduction

As one of the long-standing challenges in Natural Language Processing (NLP), Word Sense Disambiguation [Navigli, 2009, WSD] has received considerable attention over recent years. Indeed, by dealing with lexical ambiguity an effective WSD model brings numerous benefits to a variety of downstream tasks and applications, from Information Retrieval and Extraction [Zhong and Ng, 2012, Delli Bovi et al., 2015] to Machine Translation [Carpuat and Wu, 2007, Xiong and Zhang, 2014, Neale et al., 2016, Liu et al., 2017]. Recently, WSD has also been leveraged to build continuous vector representations for word senses [Chen et al., 2014, Iacobacci et al., 2015, Flekova and Gurevych, 2016].

Inasmuch as WSD is described as the task of associating words in context with the most suitable entries in a pre-defined sense inventory, WSD approaches to date can be grouped into three main categories: unsupervised, knowledge-based and supervised. Unsupervised knowledge-free approaches do not require any sense-annotated corpus nor lexical resources, inducing word senses automatically from raw corpora. Even though they suffer from data sparsity and an intrinsic difficulty in their evaluation [Agirre et al., 2006, Brody and Lapata, 2009, Manandhar et al., 2010, Van de Cruys and Apidianaki, 2011, Di Marco and Navigli, 2013, Pilehvar and Navigli, 2014], indeed, recently there is an increasing effort on making unsupervised systems more interpretable [Panchenko et al., 2017a,b]. On the other hand, knowledge-based approaches rely on the structure and content of readily-available knowledge resources. One of the first approaches of this kind was Lesk [1986], which in its original version consisted of calculating the overlap between the context of the target word and its definitions as given by the sense inventory. Based on the same principle, various works have adapted the original algorithm by also taking into account definitions from related words [Banerjee and Pedersen, 2003], or by calculating the distributional similarity between definitions and the context of the target word [Basile et al., 2014, Chen et al., 2014]. In addition to these approaches based on distributional similarity, an important branch of knowledge-based systems found their techniques on the structural properties of semantic graphs from lexical resources [Agirre and Soroa, 2009, Guo and Diab, 2010, Ponzetto and Navigli, 2010, Agirre et al., 2014, Moro et al., 2014b, Weissenborn et al., 2015, Tripodi and Pelillo, 2017]. Generally, these graph-based WSD systems first create a graph representation of the input text and then exploit different graph-based algorithms over the given representation (e.g., PageRank) to perform WSD. Lastly, supervised techniques require huge amounts of annotated data, from which extract features to train a classifier. These features have been mostly based on the information provided by the surroundings words of the target word and its collocations [Lee and Ng, 2002, Navigli, 2009].

In general the field does not have a clear path, partially owing to the fact that identifying real improvements over existing approaches becomes a hard task with current evaluation benchmarks. This is mainly due to the lack of a unified framework, which prevents direct and fair comparison among systems. Even though many evaluation datasets have been constructed for the task [Edmonds and Cotton, 2001, Snyder and Palmer, 2004, Navigli et al., 2007, Pradhan et al., 2007, Agirre et al., 2010, Navigli et al., 2013, Moro and Navigli, 2015, *inter alia*], they tend to differ in format, construction guidelines and underlying sense inventory. In fact, also a general-purpose framework for word sense disambiguation, i.e., DKPro WSD [Miller et al., 2013], that is designed to support the needs of WSD researchers, has a faq web page[1] in which they explain a way how to cope with errors and warnings given by the xml format of the dataset with various patches and conversion scripts. Moreover they also mention the issue regarding the different sense inventory and that in some cases *"you will not be able to achieve full accuracy, since some of the sense keys found in the answer key won't exist in the sense inventory."*. Indeed, in the case of the datasets annotated using WordNet [Miller, 1995], the *de facto* sense inventory for WSD, we encounter the additional barrier of having text annotated with different versions. These divergences are in the main solved individually by using or constructing automatic mappings. The quality check of such mapping, however, tends to be impractical and this leads to mapping errors which give rise to additional

[1] http://dkpro.github.io/dkpro-wsd/faq/

system inconsistencies in the experimental setting. This issue is directly extensible to the training corpora used by supervised systems. In fact, results obtained by supervised systems reported in the literature are not completely reliable, because the systems may not necessarily have been trained on the same corpus, or the corpus was preprocessed differently, or annotated with a sense inventory different from the test data. For instance, Agirre et al. [2014] note that using WordNet 3.0, instead of 1.7 or 2.1, can cause a drop in performance. Moreover, in Chaplot et al. [2015], the authors stated *"We would like to highlight some difficulties faced while calculating the exact accuracies on the datasets used for comparison."*, an issue raised up by the different version of the sense inventory of the test sets.

A clear example of what can happen is shown in Yuan et al. [2016], where the authors claim a performance increase from 5 to 10 points F-score, with respect to state of the art systems. However, their underlying model exploits proprietary data not available to the research community, and the sense inventory of the test sets is different from those used by the competitors. Thus, together, the foregoing issues prevent us from drawing reliable conclusions on different models, as in some cases ostensible improvements may have been obtained as a consequence of the nature of the training corpus, the preprocessing pipeline or the version of the underlying sense inventory, rather than the model itself. Moreover, because of these divergences, current systems tend to report results on a few datasets only, making it hard to perform a direct quantitative comparison. For instance, Basile et al. [2014] tested their system only on a recent test set, without performing an evaluation on all the previous ones.

For this reason the first focus of this thesis has been on providing to the research community a complete evaluation framework for all-words Word Sense Disambiguation overcoming all the aforementioned limitations by standardizing the WSD datasets and training corpora into a unified format, semi-automatically converting annotations from any dataset to the same version of WordNet, and preprocessing the datasets by consistently using the same pipeline. Moreover, we use this evaluation framework to perform a fair quantitative and qualitative empirical comparison of the main techniques proposed in the WSD literature.

Supervised models have been shown to outperform knowledge-based ones in standard benchmarks, at the expense, however, of harder training and limited flexibility [Navigli, 2009]. A crucial limitation of current supervised approaches is that a dedicated classifier (called *word expert*) needs to be trained for every target lemma, making them less flexible and hampering their use within end-to-end applications. In contrast, knowledge-based systems do not require sense-annotated data and often draw upon the structural properties of lexico-semantic resources [Agirre et al., 2014, Moro et al., 2014b, Weissenborn et al., 2015]. Such systems construct a model based only on the underlying resource, which is then able to handle multiple target words at the same time and disambiguate them jointly, whereas word experts are forced to treat each disambiguation target in isolation. Another key issue is multilinguality. In fact, in the last multilingual WSD competitions [Navigli et al., 2013, Moro and Navigli, 2015], in which only testing data was provided, no supervised system was submitted, because there are no available training data for languages other than English.

For these reasons, in the second part of the thesis, we depart from previous approaches and adopt a different perspective on the task: instead of framing a separate classification problem for each given word, we aim at modeling the joint disambiguation of the target text as a whole in terms of a sequence labeling problem. From this standpoint, WSD amounts to translating a sequence of words into a sequence of potentially sense-tagged tokens.With this in mind, we design, analyze and compare experimentally various neural architectures of different complexities, ranging from a single bidirectional Long Short-Term Memory [Graves and Schmidhuber, 2005, LSTM] to a sequence-to-sequence approach [Sutskever et al., 2014]. Each architecture reflects a particular way of modeling the disambiguation problem, but they all share some key features that set them apart from previous supervised approaches to WSD: they are trained end-to-end from sense-annotated text to sense labels, and learn a single all-words model from the training data, without fine tuning or explicit engineering of local features. Moreover, for the first time in WSD, to the best of our knowledge, we are able to train a system only on English data and test it on other languages, obtaining promising performance on a multilingual standard benchmark.

However, hand-labeled sense annotations are notoriously difficult to obtain on a large scale, and already available manually curated corpora [Miller et al., 1993, Passonneau et al., 2012] have a limited size. Semantically annotated corpora are indispensable in order to provide solid training and testing grounds for the development of disambiguation systems [Pilehvar and Navigli, 2014]. Indeed, encoding semantic information is a very demanding task, which can rarely be performed with high accuracy on a large scale. First of all, obtaining reliable sense-annotated corpora is highly expensive and especially difficult when non-expert annotators are involved [Lopez de

Lacalle and Agirre, 2015], and as a consequence approaches based on unlabeled data and semi-supervised learning are emerging more frequently [Taghipour and Ng, 2015b, Başkaya and Jurgens, 2016, Yuan et al., 2016, Pasini and Navigli, 2017]. Naturally, one straightforward way to obtain sense annotated data is to use a multilingual knowledge-based system to label raw text [Moro et al., 2014a] and then train a classifier. However, in order to get a better generalization of the supervised system we need to get high-quality sense-annotated data. To get better sense-annotated corpora for multiple languages, we coupled a state-of-the-art knowledge-based disambiguation system which is designed to exploit at best a multiple language setting together with a distributional similarity approach targeted at identifying a subset of sense annotations disambiguated with high confidence. Exploiting these available systems we are able to get high quality annotations for a corpus of textual definitions in multiple languages, and from parallel corpora, without relying on word alignments against a pivot language, but instead leveraging all languages at the same time in a joint disambiguation procedure that is subsequently refined using distributional similarity. Constructing a large-scale high quality sense-annotated multilingual corpus has the potential to boost both Word Sense Disambiguation and Machine Translation research [Liu et al., 2017].

Even though the annotations are proved to be of high quality, we are still exploiting off-the-shelf systems to obtain them. Over the last decade, collaborative resources like Wikipedia (an online encyclopedia) have grown not only quantitatively, but also in terms of their degree of multilingualism, i.e., the range of different languages in which they are available. In this respect, semi-structured resources [Hovy et al., 2013] stand as a convenient middle ground between high-quality, human-curated repositories and unstructured text; among others, Wikipedia constitutes an extraordinary source of semantic information for innumerable tasks in Natural Language Processing (NLP), from Named Entity Disambiguation [Cucerzan, 2007, Barrena et al., 2015] to Semantic Similarity [Gabrilovich and Markovitch, 2007, Wu and Giles, 2015] and Information Extraction [Wu and Weld, 2010]. Thus, another important goal we targeted is to augment Wikipedia with as much semantic information as possible, by recovering potentially linkable mentions not covered by original hyperlinks, with no need for recourse to an off-the-shelf disambiguation system. To achieve this, we rely only on the structure of Wikipedia itself, exploiting direct connections among Wikipedia articles and categories in order to propagate hyperlink information across the corpus. We also leverage the wide-coverage semantic network of BabelNet [Navigli and Ponzetto, 2012] and its connections across Wikipedias in different languages, as well as across different lexicographic and encyclopedic resources. As a result, we obtain and make available to the community a large sense-annotated corpus with more than 200 million annotations of over 4 million different words, covering almost 40% of the nouns in Wikipedia (compared to less than 20% covered by the original hyperlinks). In addition to confirming the quality of the annotations, we also show that our corpus constitutes a key semantic resource, leading to important new performance baselines in several tasks.

1.1 Objectives

In this thesis, we first focus our attention on studying the underlying difficulties of WSD, with the goal of facilitating the development of the task. Then we investigate supervised approaches and design neural models directly tailored to WSD while tackling the problem of the knowledge acquisition bottleneck.
The main objectives of this thesis are:

- To analyse and study the current status of the WSD task in the literature, with the aim of giving a unified representation of the data in a single standard sense inventory.
- To compare the current state of the art systems in a fair setting without using any proprietary data unavailable to the community.
- To develop a supervised approach to jointly disambiguate all words in a sentence which is flexible enough to be adapted to languages without further training.
- To develop approaches aiming to automatically generate sense-annotated corpora with high-quality annotations in multiple languages.

1.2 Contributions

This thesis provides the following significant contributions to each objectives:

- **A Unified Framework for WSD.** We present the construction of a unified framework containing all the standard test sets of the Senseval/SemEval series, reunited in a single XML format, sharing the same sense inventory (Chapter 4).

- **Empirical and fair comparison among systems.** We show and analyse the performance of the major supervised and knowledge-based systems for WSD, in a fair testing setting (Chapter 4).

- **A robust multilingual supervised system.** We put forward an approach for WSD following the sequence labelling paradigm. We conduct several experiments demonstrate that the system is statistically significance with the best system across all the test sets. Moreover, we show how to cope with the lack of training data in more languages which so far impeded the development of cross-lingual systems (Chapter 5).

- **Several methodologies for generating sense annotated data.** We present different techniques for automatically label raw corpora with high quality sense annotations. From using off-the-shelf systems to exploit at best semi-structured resource, we show how to overcome coverage limitations in multiple languages (Chapter 6).

1.3 Individual contributions

I personally contributed to the design and implementation of all the algorithms and the evaluations setup presented in this thesis, with little exceptions. In Section 6.3, I took care of the methodologies (the entire hyperlink pipeline, except for the CP heuristic), the intrinsic evaluation and the experiments on disambiguation. In Section 6.2.4, I contributed to the preprocessing and to the intrinsic evaluations, while in Section 6.2.5 to the preprocessing and the experiments.

Published material not included in this thesis. Other works, which did not contribute directly to this thesis or done before starting the Ph.D. program, and are thus not included but represent valuable effort and contribution, are, in order of publication:

- Entity Linking meets Word Sense Disambiguation: a Unified Approach [Moro et al., 2014b].

- Semantic Indexing of Multilingual Corpora and its Application on the History Domain [Raganato et al., 2016].

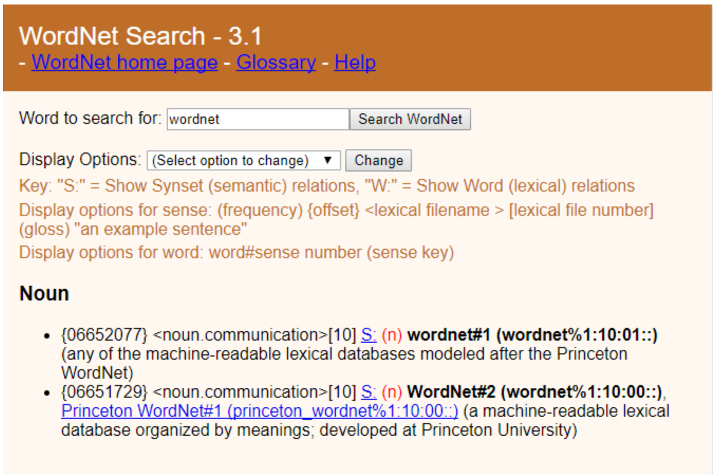
1.4 Outline of the Thesis

The thesis is organized as follows. Chapter 2 provide some preliminaries notion about the tools and the knowledge resources used across the thesis. Chapter 3 describes an overview of the literature about supervised WSD systems, explaining the difficulties tackled on this thesis. Chapter 4 gives details on how we create a unified framework for WSD, drawing a fair analysis on the performance of various systems. We then present, in Chapter 5, Seq2Sense, neural models addressing WSD as sequence labelling problem and able to seamlessly handle different languages at testing time, enabling for the first time cross-linguality. In Chapter 6 we explain how to get high quality sense annotated data, leveraging existing tools or exploiting at best semi-structured resources. Finally, Chapter 7 provides concluding remarks and highlights future works.

Chapter 2

Background: Tools and Knowledge Resources

In this chapter we provide some background information about the main resources and tools used in this work, namely WordNet, Wikipedia, BabelNet, Babelfy and NASARI.



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: wordnet

Display Options: (Select option to change) ▼

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (frequency) {offset} <lexical filename > [lexical file number] (gloss) "an example sentence"
Display options for word: word#sense number (sense key)

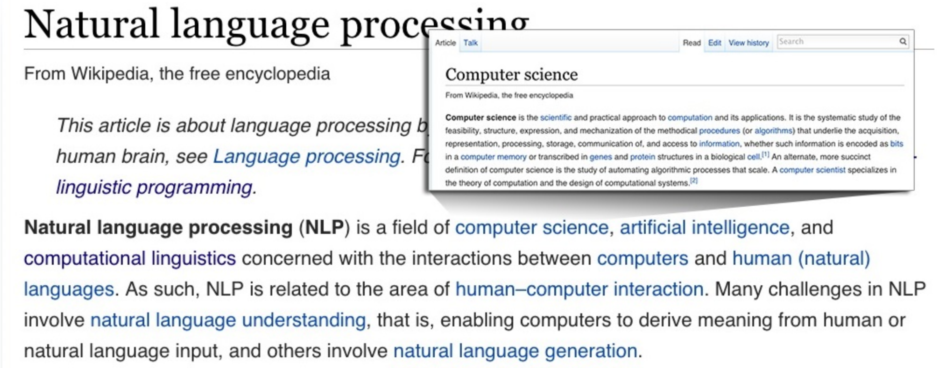
Noun

- {06652077} <noun.communication>[10] S: (n) **wordnet#1 (wordnet%1:10:01::)** (any of the machine-readable lexical databases modeled after the Princeton WordNet)
- {06651729} <noun.communication>[10] S: (n) **WordNet#2 (wordnet%1:10:00::)**, Princeton WordNet#1 (princeton_wordnet%1:10:00::) (a machine-readable lexical database organized by meanings; developed at Princeton University)

Figure 2.1. WordNet definitions by WordNet itself.

WordNet. The Princeton WordNet of English [Miller, 1995] is by far the most widely used computational lexicon in Natural Language Processing. It is manually curated by expert lexicographers and organized as a semantic network, where concepts are connected via lexico-semantic relations. Its internal structure is based on synset, i.e., words with the same meaning grouped together. Similarly to traditional dictionaries, WordNet provides a textual definition (*gloss*), as well as small usage examples for each synset. Being hand-crafted by expert annotators, definitional knowledge from WordNet is among the most accurate available and includes also non-nominal parts of speech rarely covered by other resources (e.g., adjectives and adverbs). All over the years, WordNet has been used for innumerable tasks, however, being a lexicographic network, it provides definitions only for concepts missing named entities at all.

Wikipedia. Wikipedia[1] is a well-known freely available collaborative encyclopedia, containing 40 million pages in over 299 languages. The Wikipedia internal links (see Figure 2.2) are one of the features that makes Wikipedia a valuable project and resource. In fact it was estimated that the network of internal links offers the opportunity to proceed from any article to any other with an average of 4.5 clicks [Dolan, 2008].



Natural language processing

From Wikipedia, the free encyclopedia

This article is about language processing b human brain, see Language processing. F linguistic programming.

Natural language processing (**NLP**) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation.

Computer science

From Wikipedia, the free encyclopedia

Computer science is the scientific and practical approach to computation and its applications. It is the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information, whether such information is encoded as bits in a computer memory or transcribed in genes and protein structures in a biological cell.[1] An alternate, more succinct definition of computer science is the study of automating algorithmic processes that scale. A computer scientist specializes in the theory of computation and the design of computational systems.[2]

Figure 2.2. A sample Wikipedia page with links.

The freedom to create and edit pages has a positive impact both qualitatively and quantitatively, matching and overcoming the famous *Encyclopedia Britannica* [Giles, 2005]. It was estimated that the text of the English Wikipedia is currently equivalent to over 2000 volumes of the *Encyclopedia Britannica*[2].

Wikipedia users are free to create new pages following the guidelines provided by the encyclopedia. In fact, each article in Wikipedia is identified by a unique identifier allowing the creation of shortcuts, expressed as: [[ID |anchor text]], where the anchor text is the fragment of text of a page linked to the identified page ID, and [[anchor text]], where the anchor text is linked to the corresponding homonymous

[1] http://www.wikipedia.org

[2] `http://en.wikipedia.org/wiki/Wikipedia:Size_in_volumes`

page.
For instance, in the following sentence taken from the Wikipedia page Natural Language Processing: *"Natural language processing (NLP) is a field of [[computer science]], [[artificial intelligence]], and [[computational linguistics]] concerned with the interactions between [[computer]]s and [[Natural language\human (natural) languages]]. As such, NLP is related to the area of [[human-computer interaction]]. Many challenges in NLP involve [[natural language understanding]], that is, enabling computers to derive meaning from human or natural language input, and others involve [[natural language generation]].",* the users decided to link *human (natural) languages* to the Wikipedia page *Natural language*.
Today, Wikipedia represents an extraordinary resource in Natural Language Processing [Cucerzan, 2007, Gabrilovich and Markovitch, 2007, Wu and Weld, 2010, Chen et al., 2017a]. Due to its focus on encyclopedic knowledge, Wikipedia contains almost exclusively nominal senses (such as named entities or specialized concepts).

BabelNet. BabelNet [Navigli and Ponzetto, 2012] is a large-scale, multilingual encyclopedic dictionary (i.e., a resource where both lexicographic and encyclopedic knowledge is available in multiple languages) obtained from the automatic integration of heterogeneous resources such as WordNet, Open Multilingual WordNet [Bond and Foster, 2013], Wikipedia[3], OmegaWiki[4], Wiktionary[5], Wikidata[6], Wikiquote[7], VerbNet [Kipper et al., 2008], Microsoft Terminology[8], GeoNames[9], WoNeF [Pradet et al., 2014], ItalWordNet [Roventini et al., 2000], ImageNet [Deng et al., 2009] and FrameNet [Baker et al., 1998]. The integration is performed via an automatic mapping between these resources which result in merging equivalent concepts from the different resources. BabelNet covers and links named entities and concepts present in all the aforementioned resources obtaining a wide coverage resource containing both lexicographic and encyclopedic terms. Each concept or entity inside BabelNet is associated with a synonym set, called Babel synset, comprising lexicalizations and glosses of that concept or entity in a variety of languages, interconnected with several semantic relations.

[3] `http://www.wikipedia.org`

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

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

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

[7] `http://www.wikiquote.org/`

[8] `http://www.microsoft.com/Language/en-US/Terminology.aspx`

[9] `http://www.geonames.org/`

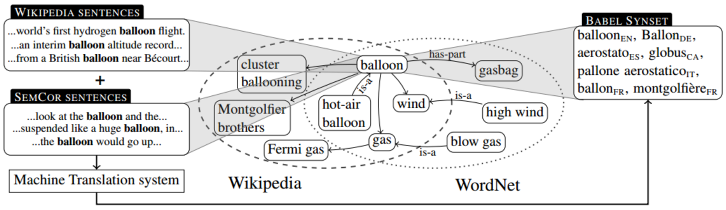
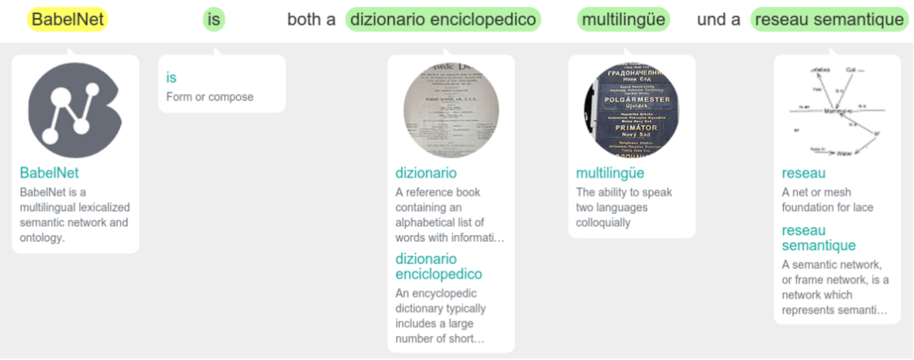

WIKIPEDIA SENTENCES
...world's first hydrogen **balloon** flight.
...an interim **balloon** altitude record...
...from a British **balloon** near Bécourt...
+
SEMCOR SENTENCES
...look at the **balloon** and the...
...suspended like a huge **balloon**, in...
...the **balloon** would go up...
Machine Translation system
cluster ballooning
balloon
has-part
gasbag
is-a
Montgolfier brothers
hot-air balloon
wind
is-a
high wind
gas
Fermi gas
is-a
blow gas
Wikipedia
WordNet
BABEL SYNSET
balloon$_{EN}$, Ballon$_{DE}$,
aerostato$_{ES}$, globus$_{CA}$,
pallone aerostatico$_{IT}$,
ballon$_{FR}$, montgolfière$_{FR}$

Figure 2.3. An illustrative overview of BabelNet (picture from Navigli and Ponzetto [2012]).

For instance in Figure 2.3 the concepts *balloon, wind, hot-air balloon and gas* are defined in both Wikipedia and WordNet while *Montgolfier brothers* and *blow gas* are respectively named entities and concepts retrieved from Wikipedia and WordNet. The latest release of BabelNet, i.e., 3.7, has now become the largest resource of its kind, providing a full-fledged taxonomy [Flati et al., 2016], covering 271 languages with more than 13M Babel synsets and 380M lexico-semantic relations (for more statistics see `http://babelnet.org/stats`). It is also available as SPARQL endpoint and in RDF format containing up to 2 billion RDF triples.

Babelfy. Babelfy [Moro et al., 2014b] is a graph-based approach to joint multilingual Entity Linking and Word Sense Disambiguation, a state-of-the-art system in both tasks. Babelfy is based on the BabelNet semantic network and jointly performs disambiguation in three steps. The first step associates with each node of the network a set of semantically relevant vertices, i.e., concepts and named entities, thanks to a notion of semantic signatures. This is a preliminary step which needs to be performed only once, independently of the input text. The second step extracts all the textual mentions from the input text, i.e., substrings of text for which at least one candidate named entity or concept can be found in BabelNet. Consequently, for each extracted mention, it obtains a list of the possible meanings according to the semantic network. The last step consists of connecting the candidate meanings according to the previously-computed semantic signatures. It then extracts a dense sub-graph and selects the best candidate meaning for each fragment.
Being language-independent, the algorithm can easily be applied to any language for which lexicalizations are available inside the underlying semantic network. As a result, Babelfy can handle mixed text in which multiple languages are used at the same time, or even work without being supplied with information as to which languages the input text contains (*language-agnostic* setting) (see Figure 2.4).



BabelNet is both a dizionario enciclopedico multilingüe und a reseau semantique

BabelNet
BabelNet is a multilingual lexicalized semantic network and ontology.

is
Form or compose

dizionario
A reference book containing an alphabetical list of words with informati…

dizionario enciclopedico
An encyclopedic dictionary typically includes a large number of short…

multilingüe
The ability to speak two languages colloquially

reseau
A net or mesh foundation for lace

reseau semantique
A semantic network, or frame network, is a network which represents semanti…

Figure 2.4. Output of the Babelfy system on a code-switching sentence.

NASARI. NASARI [Camacho-Collados et al., 2016] is a vectorial representation of concepts and entities from the BabelNet sense inventory. NASARI has proved to be effective in various NLP tasks, such as semantic similarity and WSD [Shalaby and Zadrozny, 2015, Camacho-Collados et al., 2016, Tripodi and Pelillo, 2017], knowledge-base construction and alignment [Lieto et al., 2016, Espinosa Anke et al., 2016, Camacho-Collados and Navigli, 2017, Cocos et al., 2017], object recognition [Young et al., 2016] and text classification [Pilehvar et al., 2017]. NASARI leverages structural properties from BabelNet and word embeddings trained on large corpora. Given a Babel synset, its NASARI representation is computed by first gathering a relevant sub-corpus of contextual information from Wikipedia, exploiting both the Wikipedia inter-link structure and the BabelNet taxonomy. All content words in this sub-corpus are then tokenized, lemmatized and weighted using *lexical specificity* [Lafon, 1980], a statistical measure based on the hypergeometric distribution that measures the relevance of a word in a given sub-corpus[10]. Finally, the sub-corpus is turned into a vector using three different techniques that give rise to three different types of representation: *lexical*, *unified*, and *embedded*. In this thesis we rely on the latter type (NASARI-*embed*). The word embeddings used for NASARI-embed are the pre-trained vectors of Word2Vec [Mikolov et al., 2013a], trained on the Google News corpus. These 300-dimensional word embeddings are injected into the NASARI embedded representation via a weighted average, where the weights are given by lexical specificity. The resulting vector is still defined at the sense level, but lies in the same semantic space as word embeddings, thus enabling a direct comparison between words and synsets.

[10] Lexical specificity has been shown to outperform *tf-idf* as a vector weighting scheme [Camacho-Collados et al., 2015a].

Chapter 3

Supervised Word Sense Disambiguation: How far have we come?

The literature on WSD is broad and comprehensive [Agirre and Edmonds, 2007, Navigli, 2009], in this chapter our focus is on the supervised one. From the classical machine learning tools such as Decisions Lists and Trees, Naive Bayes classifiers, a lot has been made from the NLP community. Over the last decade, we have been witnessed a real upsurge of machine learning models in the NLP community, specially exploiting neural networks and deep learning. Traditional approaches are generally based on extracting local features from the words surrounding the target, and then training a classifier [Zhong and Ng, 2010, Shen et al., 2013] for each target lemma, calling this paradigm word expert. Usually, the classifier is concerned with a single word and performs a classification task in order to assign the appropriate sense to each instance of that word. Now, we will give a brief overview of the most popular machine learning systems tackling the WSD task.

IMS. It Makes Sense (i.e., IMS) [Zhong and Ng, 2010], uses a Support Vector Machine (SVM) classifier over a set of conventional WSD features. The default implementation includes surrounding words, part of speech tags of surroundings words, and local collocations as features. IMS makes extensive use of the basis of a NLP pipeline. Detecting the sentence boundaries in a raw input text with a sentence splitter. Tokenizing the sentence. Assigning part of speech tags to all tokens with a PoS tagger. Finally, finding the lemma form of each token with a lemmatizer. As a final step, the IMS system exploits the scored output of the classifier to select the word sense. Nowadays, IMS represents a hard baseline to beat, as concerns supervised approach. The major drawback of the system is that being feature-based, it needs preprocessing tools, plus, currently, there are no studies on how much these features are helpful for other languages rather than English.

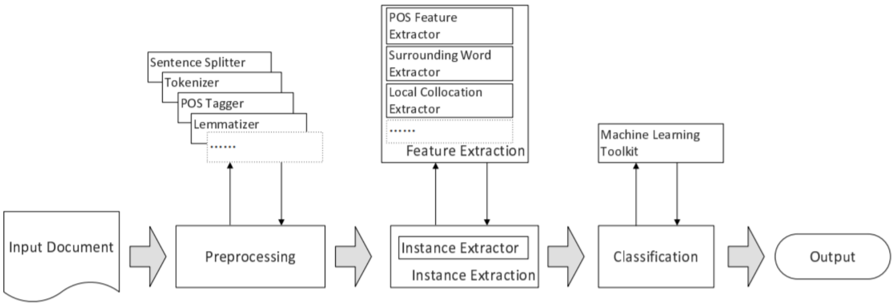


Figure 3.1. IMS architecture (picture from Zhong and Ng [2010]).

Recently, more complex features based on word embeddings trained on unlabeled corpora have also been explored. These approaches have shown the potential of using word embeddings on the WSD task [Taghipour and Ng, 2015b, Rothe and Schütze, 2015, Iacobacci et al., 2016].

IMS+embeddings. Iacobacci et al. [2016] carried out a comparison of different strategies for integrating word embeddings as a feature in WSD to train a linear classifier. The first method concatenates the vectors of the words surrounding the target word as in Bengio et al. [2003]. The second one, computes the centroid of the embeddings of all the surrounding words. In the third and fourth method surrounding words are weighted based on their distance from the target word, weighting the vectors inversely proportional to their distance from the target and exponentially respectively. Integrating the last strategy together with the default features of IMS proved to achieve the best performance overall. The authors tested pre-trained embeddings such as Word2vec [Mikolov et al., 2013a] trained on the Google News corpus with 300 dimensions, the 300 dimensional embeddings of GloVe [Pennington et al., 2014], the 50 dimensional C&W embeddings [Collobert and Weston, 2008], and a PMI-SVD vector space model trained by Baroni et al. [2014]. In addition they studied different configurations, from the dimensionality (200, 400, or 800) of the embeddings, combination strategy, window size (5, 10, 20 and words), and the standard WSD features (collocations, POS tags, surrounding words, all of these or none). The best parameters was achieved by the Skip-gram model of Word2Vec with 400 as dimension, 10 negative sampling, window size of 10 and sub-sampling of frequent words to 10^{-3}, with the exclusion of the surrounding words as feature.

However, the publicly available implementations of IMS and IMS+embeddings suffer from several drawbacks: the design of the software makes the current code difficult to extend (e.g., with classes taking as input more than 15 parameters), the implementation is not optimized for large datasets, being rather time- and resource-consuming. These difficulties hamper the work of contributors willing to update it, as well as the effort of researchers that would like to use it with languages other than English. For this reason we developed SUPWSD, whose objective is to overcome the aforementioned drawbacks, and facilitate the use of supervised WSD software for both end users and researchers. More details about the software are given in the Appendix.

Even though word embeddings provide a good boost to the performance when integrated as features, the underline model remains IMS with the same aforementioned drawbacks. The recent upsurge of neural networks has also contributed to fueling WSD research. Starting to remove engineered features from the model, training end-to-end classifier for the task.

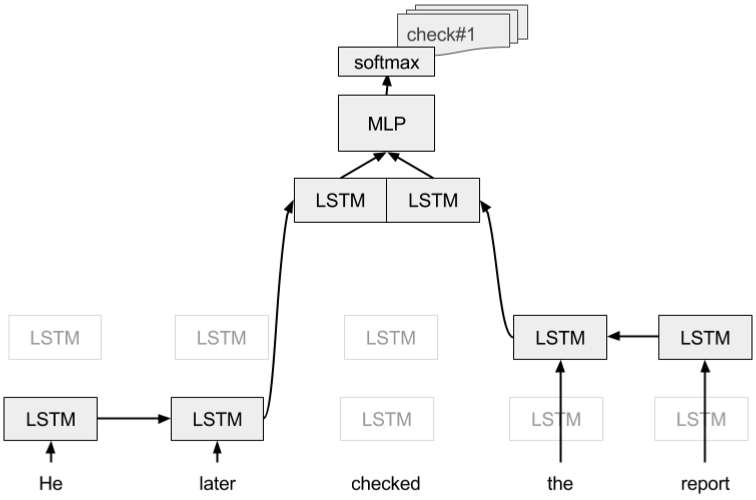


Figure 3.2. The architecture of Kågebäck and Salomonsson [2016].

Kågebäck and Salomonsson. Kågebäck and Salomonsson [2016] trained a bidirectional LSTM directly tailored to WSD (Figure 3.2). The model being based on an LSTM is able to take into account word order when classifying, relying on no language specific features. The architecture consists of an embedding layer, a Bidirectional LSTM, a hidden layer and a softmax layer. The system center the target word and computes a probability distribution over the possible candidate senses of the word itself. Moreover, the authors introduced a regularization technique, called DropWord, (similar to word dropout [Srivastava et al., 2014]) in which the word to be dropped is replaced with a specific tag <*dropped*>, corresponding to a new word embedding to be trained. In their experiments, the authors used pre-trained embeddings, the GloVe vectors [Pennington et al., 2014] trained on Wikipedia and Gigaword with 200 dimension, two LSTMs of size 74 and a hidden layer of size 200. This system has been evaluated only on the English lexical sample WSD tasks [Kilgarriff, 2001, Mihalcea et al., 2004] proving, despite its simplicity, to reach good results in line with state-of-the-art systems.

Thanks to the trend of deep learning and a better encoding of context and sentence representation, recently there has been a shifting towards systems instance based. Instance learning is a supervised method in which the classification model is built from examples, a typical approach is the k-Nearest Neighbor (kNN) algorithm. Neural language models have shown their potential in this respect [Melamud et al., 2016, Yuan et al., 2016].

Context2Vec. Melamud et al. [2016] use a bidirectional LSTM to learn a context embedding from large corpora. The model learn a generic embedding function for the context around a target word. First they fed the words into two LSTMs, concatenating the vectors. The concatenation is given in input to a multi-layer perceptron (MLP) output an embedding representing the context around the target word of the same dimension of the embedding of the target word itself. To learn the parameters of the network, the authors used Word2Vec's negative sampling objective function. The authors used two LSTMs of 600 as dimension, 1200 hidden units for the MLP and 600 for the context embedding. The system has been tested on the Microsoft Sentence Completion Challenge [Zweig and Burges, 2011], on the Lexical Substitution tasks [McCarthy and Navigli, 2007, Kremer et al., 2014] and on the most recent English lexical sample WSD task [Mihalcea et al., 2004]. As

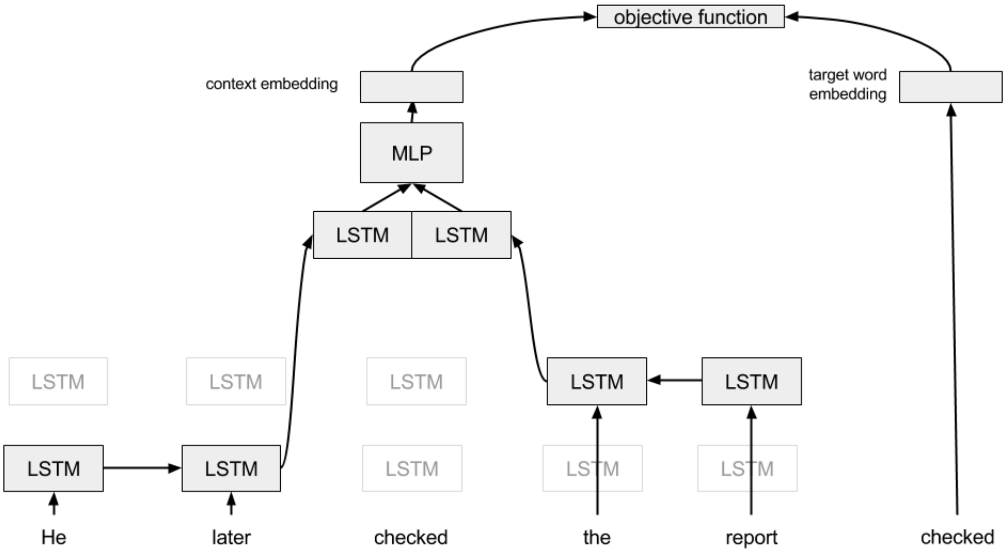


Figure 3.3. Context2Vec architecture.

baseline, they used word embeddings trained with the popular Word2Vec Skip-gram model, representing the context as a simple average of the embedding of the words inside the sentence. As regards the WSD task, a context vector is learned for each sense annotation in the training corpus and the sense annotation whose context vector is closer to the target word's context vector is selected as the intended sense. This approach beat the baseline on all experiments.

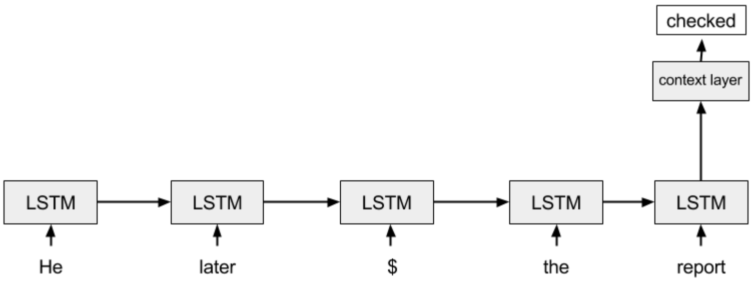


Figure 3.4. The LSTM language model of Yuan et al. [2016]. The word to predict is replaced by a special symbol $ and predicted at the end of the sentence.

Yuan et al. Similar to Context2Vec, Yuan et al. [2016] train a LSTM language model to predict a word given the surrounding context, on a big unlabelled corpus (see Figure 3.4). From the training sentences they compute sense vectors, averaging the context vectors of the LSTM of the same sense. Then, the algorithm classify a word in a context by finding the sense vector with the maximum cosine similarity to the context vector of the word to label in the test sentence. The authors used a LSTM of 2048 units, a 512 dimensional context layer and 512 dimensional word embeddings trained on a 100 billion word news corpus. Moreover, to augment the training data, the authors present a label propagation method to annotated a large number of unlabeled sentences from the web [Talukdar and Crammer, 2009]. The system was tested on five different WSD test set [Edmonds and Cotton, 2001, Snyder and Palmer, 2004, Navigli et al., 2007, Pradhan et al., 2007, Navigli et al., 2013], outperforming the state-of-the-art (i.e., IMS+embeddings) in all benchmarks.

In this thesis we compare supervised systems and study the role of their underlying sense-annotated training corpus. Since semi-supervised models have been shown to outperform fully supervised systems in some settings [Taghipour and Ng, 2015b, Başkaya and Jurgens, 2016, Iacobacci et al., 2016, Yuan et al., 2016], we evaluate and compare models using both manually-curated and automatically-constructed sense-annotated corpora for training.

All these contributions have shown that supervised neural models can achieve state-of-the-art performances without taking advantage of external resources or language-specific features. However, they all consider each target word as a separate classification problem and, to the best of our knowledge, very few attempts have been made to disambiguate a text jointly using sequence learning [Ciaramita and Altun, 2006]. Sequence learning, especially using LSTM [Hochreiter and Schmidhuber, 1997, Graves and Schmidhuber, 2005, Graves, 2013], has become a well-established standard in numerous NLP tasks [Zhou and Xu, 2015, Ma and Hovy, 2016, Wang and Chang, 2016]. In particular, sequence-to-sequence models [Sutskever et al., 2014] have grown increasingly popular and are used extensively in, e.g., Machine Translation [Cho et al., 2014, Bahdanau et al., 2015], Sentence Representation [Kiros et al., 2015], Syntactic Parsing [Vinyals et al., 2015], Conversation Modeling [Vinyals and Le, 2015], Morphological Inflection [Faruqui et al., 2016] and Text Summarization [Gu et al., 2016]. In line with this trend, we focus on the (so far unexplored) context of supervised WSD, and investigate state-of-the-art all-words approaches that are based on neural sequence learning and capable of disambiguating all target content words within an input text, a key feature in several knowledge-based approaches. Moreover, we investigated how to adapt a supervised WSD model also for languages without any training data. To the best of our knowledge, we are the first to explore the potential of a neural system trained in a language and tested on another one targeting the WSD task (see Chapter 5).

However, all these contributions are trained and tested on small sense annotated corpora where most of senses of the underlying sense inventory lack of annotations. These systems could become worthless in downstream application where most of the words are simply not covered in the training data, so missing to annotate them. In general, the drawback of using supervised models arises from the so-called *knowledge-acquisition bottleneck*, a problem that becomes particularly vexed when such models are applied to larger inventories, due to the vast amount of annotated data they normally require. This is mainly due to the expensive manual effort required to annotate large corpora.

Over the years, the WSD community has created a range of different sense-annotated datasets for a variety of evaluation tasks. A well-known example for WSD is the Senseval/SemEval competition series [Edmonds and Cotton, 2001, Snyder and Palmer, 2004, Navigli et al., 2007, Pradhan et al., 2007, Agirre et al., 2010, Navigli et al., 2013, Moro and Navigli, 2015], where manually annotated datasets are released. The largest dataset manually annotated with word senses is SemCor [Miller et al., 1993], a subset of the English Brown Corpus, with more than 200K content words tagged using the WordNet lexical database. Nevertheless, many instances of SemCor have very few annotations and only a small set of polysemous words is well covered. To bridge this gap, various automatic methods have been developed to generate training data on a larger scale, from unsupervised bootstrapping [Diab, 2004], to word alignments on parallel corpora [Zhong and Ng, 2009]. More recently, Taghipour and Ng [2015a] applied the latter approach to the MultiUN corpus and obtained one million training instances, which they released as the largest publicly available dataset for WSD. Another disambiguation task focused on the coverage of the sense inventory was presented as part of the Senseval-3 workshop [Litkowski, 2004] on WordNet glosses. In fact the Princeton WordNet Gloss Corpus[1], with more than 300K manual annotations, has already been shown to be successful as part of the pipeline in semantic similarity [Pilehvar et al., 2013], domain labeling [González et al., 2012] and Word Sense Disambiguation [Agirre and Soroa, 2009, Camacho-Collados et al., 2015b] systems. However, the best reported system obtained precision and recall figures below 70%, which

[1] http://wordnet.princeton.edu/glosstag.shtml

is arguably not enough to provide high-quality sense-annotated data for current state-of-the-art NLP systems. Moreover, as new encyclopedic knowledge about the world is constantly being harvested, keeping up using only human annotation is becoming an increasingly expensive endeavor, specially if we want to scale up to multiple languages. Despite the fact that sense-annotated corpora for a number of languages have been around for more than a decade [Petrolito and Bond, 2014], they either include few samples per word sense, or only cover a restricted set of ambiguous words [Passonneau et al., 2012]; as a result, multilingual WSD was until recently almost exclusively tackled using knowledge-based approaches [Agirre et al., 2014, Moro et al., 2014b]. Nowadays, the rapid development of NLP pipelines for languages other than English has been opening up the possibilities for the automatic generation of multilingual sense-annotated data. At the same time, the prominent role of collaborative resources [Hovy et al., 2013], available in multiple languages, has created a convenient development ground for NLP systems. By bridging the gap between lexicographic and encyclopedic knowledge, BabelNet [Navigli and Ponzetto, 2012] is a key milestone in this respect. Using BabelNet, a unified multilingual sense inventory, we can obtain language-independent sense annotations for a wide variety of concepts and named entities, which can be seamlessly mapped to individual semantic resources (e.g WordNet, Wikipedia, DBpedia) via BabelNet's inter-resource mappings. With the aim of overcoming the aforementioned shortfall (i.e., getting high quality annotations while at the same time covering as much as possible the sense inventory for multiple languages), we propose an automatic disambiguation approach which leverages multilinguality and cross-resource information along with a state-of-the-art graph-based disambiguation system [Moro et al., 2014b] and a distributional representation of concepts and entities [Camacho-Collados et al., 2015a]. By exploiting at best all these components, we started disambiguating the BabelNet glosses, over 40 million definitions for 271 languages. However, glosses are limited to short and concise text, plus often they are not well syntactically structured. For this reason, we turn our attention also to parallel corpora, exploiting at best the cross-language complementarities of the translations.

Apart from leveraging off-the-shelf systems, we could exploit semi-structured resources to get high quality annotations. In this respect Wikipedia, as one of the most popular semi-structured resources in the field, provides a convenient bridge to multilinguality, with several million inter-language links among articles referring to the same concept or entity. Regardless of whether Wikipedia is seen as a multilingual semantic network of concepts and entities or as a sense-annotated corpus, hyperlinks (inter-page links) constitute its key structural property. Yet only a small fraction of mentions across the entire Wikipedia corpus is linked. The specific task of detecting and annotating potentially linkable mentions in Wikipedia has been addressed in various ways, including gamification approaches [West et al., 2015] and classifiers with Wikipedia-specific features [Noraset et al., 2014]. Instead, we do not rely on human intervention at all, nor do we utilize a trained and tuned learning system, our pipeline is fully automatic and based solely on the structure of Wikipedia able to triple the overall number of linked mentions present in Wikipedia.

To conclude this chapter, summing up what has been done by the community, word embeddings first, and recurrent neural network later proved to achieve better performance, but they are still limited to the word expert paradigm, without considering to jointly label all senses in a sentence, that a tagged sense can help to disambiguate another one in the same sentence. Moreover, we need to start to have a look also to other languages than English, because so far no supervised system has been tested on a multilingual level, and each language has its own difficulties to address (e.g., morphologically rich languages, different level of ambiguity, etc.) and need to be investigated. A reason why supervised all words WSD is behind, respect to other applied NLP tasks, could be found on the limited sense annotated data available, and with their inconsistencies. We are far behind from having a project like the Universal Dependencies [Nivre et al., 2016] targeting word senses, but this thesis represents a first step towards that direction.

Chapter 4

Word Sense Disambiguation: a Unified Evaluation Framework and Empirical Comparison

Research on Word Sense Disambiguation (WSD) has been hampered by the fact that all the available test sets vary across years, as a consequence it is arduous performing comparisons across systems. With different sense inventories, format and construction guidelines, it is difficult to have a fair comparison among systems, making hard to understand why a system performs better than another one. In this chapter our goal is to tackle this problem by unifying in a single format and sense inventory the most common training a testing set aiming at using this evaluation framework to perform a fair quantitative and qualitative empirical comparison.

The rest of this chapter is organized as follows. In Section 4.1, we explain our pipeline to standardize the different datasets into a unified format, semi-automatically converting annotations from any dataset to the same version of WordNet, i.e., 3.0, and by preprocessing the datasets using consistently the same pipeline. In section 4.2, we give details about the datasets took into account showing some statistics. We then use this evaluation framework to perform a fair quantitative and qualitative empirical comparison of the main techniques proposed in the WSD literature in Section 4.3, finally, we provide some analysis of the results and the concluding remarks in Sections 4.3.3 and 4.4, respectively.

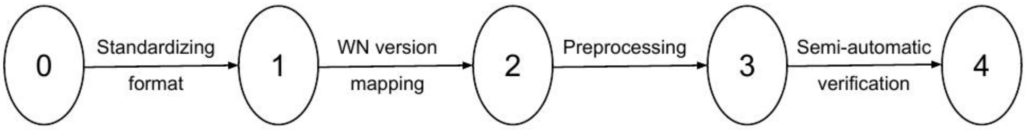


Figure 4.1. Pipeline for standardizing any given WSD dataset.

4.1 Standardization of WSD datasets

In this section we explain our pipeline for transforming any given evaluation dataset or sense-annotated corpus into a preprocessed unified format. In our pipeline we do not make any distinction between evaluation datasets and sense-annotated training corpora, as the pipeline can be applied equally to both types. For simplicity we will refer to both evaluation datasets and training corpora as WSD datasets.

Figure 4.1 summarizes our pipeline to standardize a WSD dataset. The process consists of four steps:

1. Most WSD datasets in the literature use a similar XML format, but they have some divergences on how to encode the information. For instance, the SemEval-15 dataset [Moro and Navigli, 2015] was developed for both WSD and Entity Linking and its format was especially designed for this latter task. Therefore, we decided to convert all datasets to a unified format. As unified format we use the XML scheme used for the SemEval-13 all-words WSD task [Navigli et al., 2013], where preprocessing information of a given corpus is also encoded.

2. Once the dataset is converted to a unified format, we map the sense annotations from its original WordNet version to 3.0, which is the latest version of WordNet used in evaluation datasets. This mapping is carried out semi-automatically. First, we use automatically-constructed WordNet mappings[1] [Daude et al., 2003]. These mappings provide confidence values which we use to initially map senses whose mapping confidence is 100%. Then, the annotations of the remaining senses are manually checked, and re-annotated or removed whenever necessary[2]. Additionally, in this step we decided to remove all annotations of auxiliary verbs, following the annotation guidelines of the latest WSD datasets.

[1] http://nlp.lsi.upc.edu/tools/download-map.php

[2] This manual correction involved less than 10% of all instances for the datasets for which this step was performed.

3. The third step consists of preprocessing the given dataset. We used the Stanford CoreNLP toolkit [Manning et al., 2014] for Part-of-Speech (PoS) tagging[3] and lemmatization. This step is performed in order to ensure that all systems use the same preprocessed data.

4. Finally, we developed a script to check that the final dataset conforms to the aforementioned guidelines. In this final verification we also ensured that the sense annotations match the lemma and the PoS tag provided by Stanford CoreNLP by automatically fixing all divergences.

4.2 Data

In this section we summarize the WSD datasets used in the evaluation framework. To all these datasets we apply the standardization pipeline described in Section 4.1. First, we enumerate all the datasets used for the evaluation (Section 4.2.1). Second, we describe the sense-annotated corpora used for training (Section 4.2.2). Finally, we show some relevant statistics extracted from these resources (Section 4.2.3).

4.2.1 WSD evaluation datasets

For our evaluation framework we considered five standard all-words fine-grained WSD datasets from the Senseval and SemEval competitions:

- **Senseval-2** [Edmonds and Cotton, 2001]. This dataset was originally annotated with WordNet 1.7. After standardization, it consists of 2282 sense annotations, including nouns, verbs, adverbs and adjectives.

- **Senseval-3 task 1** [Snyder and Palmer, 2004]. The WordNet version of this dataset was 1.7.1. It consists of three documents from three different domains (editorial, news story and fiction), totaling 1850 sense annotations.

- **SemEval-07 task 17** [Pradhan et al., 2007]. This is the smallest among the five datasets, containing 455 sense annotations for nouns and verbs only. It was originally annotated using WordNet 2.1 sense inventory.

[3] In order to have a standard format which may be used by languages other than English, we provide coarse-grained PoS tags as given by the universal PoS tagset [Petrov et al., 2011].

- **SemEval-13 task 12** [Navigli et al., 2013]. This dataset includes thirteen documents from various domains. In this case the original sense inventory was WordNet 3.0, which is the same as the one that we use for all datasets. The number of sense annotations is 1644, although only nouns are considered.

- **SemEval-15 task 13** [Moro and Navigli, 2015]. This is the most recent WSD dataset available to date, annotated with WordNet 3.0. It consists of 1022 sense annotations in four documents coming from three heterogeneous domains: biomedical, mathematics/computing and social issues.

4.2.2 Sense-annotated training corpora

We now describe the two WordNet sense-annotated corpora used for training the supervised systems in our evaluation framework:

- **SemCor** [Miller et al., 1993]. SemCor[4] is a manually sense-annotated corpus divided into 352 documents for a total of 226,040 sense annotations. It was originally tagged with senses from the WordNet 1.4 sense inventory. SemCor is, to our knowledge, the largest corpus manually annotated with WordNet senses, and is the main corpus used in the literature to train supervised WSD systems [Agirre et al., 2010, Zhong and Ng, 2010].

- **OMSTI** [Taghipour and Ng, 2015a]. OMSTI (*One Million Sense-Tagged Instances*) is a large corpus annotated with senses from the WordNet 3.0 inventory. It was automatically constructed by using an alignment-based WSD approach [Chan and Ng, 2005] on a large English-Chinese parallel corpus [Eisele and Chen, 2010, MultiUN corpus]. OMSTI[5] has already shown its potential as a training corpus by improving the performance of supervised systems which add it to existing training data [Taghipour and Ng, 2015a, Iacobacci et al., 2016].

[4] We downloaded the SemCor 3.0 version at `web.eecs.umich.edu/~mihalcea/downloads.html`

[5] In this thesis we refer to the portion of sense-annotated data from the MultiUN corpus as OMSTI. Note that OMSTI was released along with SemCor.

	#Docs	#Sents	#Tokens	#Annotations	#Sense types	#Word types	Ambiguity
Senseval-2	3	242	5,766	2,282	1,335	1,093	5.4
Senseval-3	3	352	5,541	1,850	1,167	977	6.8
SemEval-07	3	135	3,201	455	375	330	8.5
SemEval-13	13	306	8,391	1,644	827	751	4.9
SemEval-15	4	138	2,604	1,022	659	512	5.5
SemCor	352	37,176	802,443	226,036	33,362	22,436	6.8
OMSTI	-	813,798	30,441,386	911,134	3,730	1,149	8.9

Table 4.1. Statistics of the WSD datasets used in the evaluation framework (after standardization).

4.2.3 Statistics

Table 4.1 shows some statistics[6] of the WSD datasets and training corpora which we use in the evaluation framework. The number of sense annotations varies across datasets, ranging from 455 annotations in the SemEval-07 dataset, to 2,282 annotations in the Senseval-2 dataset. As regards sense-annotated corpora, OMSTI is made up of almost 1M sense annotations, a considerable increase over the number of sense annotations of SemCor. However, SemCor is much more balanced in terms of unique senses covered (3,730 covered by OMSTI in contrast to over 33K covered by SemCor). Additionally, while OMSTI was constructed automatically, SemCor was manually built and, hence, its quality is expected to be higher.

Finally, we calculated the ambiguity level of each dataset, computed as the total number of candidate senses (i.e., senses sharing the surface form of the target word) divided by the number of sense annotations. The highest ambiguity is found on OMSTI, which, despite being constructed automatically, contains a high coverage of ambiguous words. As far as the evaluation competition datasets are concerned, the ambiguity may give a hint as to how difficult a given dataset may be. In this case, SemEval-07 displays the highest ambiguity level among all evaluation datasets.

4.3 Evaluation

The evaluation framework consists of the WSD evaluation datasets described in Section 4.2.1. In this section we use this framework to perform an empirical comparison among a set of heterogeneous WSD systems. The systems used in the

[6] Statistics included in Table 4.1: number of documents (#Docs), sentences (#Sents), tokens (#Tokens), sense annotations (#Annotations), sense types covered (#Sense types), annotated lemma types covered (#Word types), and ambiguity level (Ambiguity). There was no document information in the OMSTI data released by Taghipour and Ng [2015a].

evaluation are described in detail in Section 4.3.1, the results are shown in Section 4.3.2 and a detailed analysis is presented in Section 4.3.3.

4.3.1 Comparison systems

We include three supervised (Section 4.3.1) and three knowledge-based (Section 4.3.1) all-words WSD systems in our empirical comparison.

Supervised

To ensure a fair comparison, all supervised systems use the same corpus for training: SemCor and SemCor+OMSTI[7] (see Section 4.2.2). Moreover, we included the Most Frequent Sense (**MFS**) heuristic as baseline, which for each target word selects the sense occurring the highest number of times in the training corpus. For a description of the supervised WSD systems used in the evaluation, see chapter 3. As concerns **IMS+emb**, in this chapter we consider the two best configurations in Iacobacci et al. [2016][8]: using all IMS default features including and excluding surrounding words (IMS+emb and IMS$_{\text{-s}}$+emb, respectively). In both cases word embeddings are integrated using exponential decay (i.e., word weights drop exponentially as the distance towards the target word increases). Likewise, we use Iacobacci et al.'s suggested learning strategy and hyperparameters to train the word embeddings: Skip-gram model of Word2Vec[9] [Mikolov et al., 2013a] with 400 dimensions, ten negative samples and a window size of ten words. As unlabeled corpus to train the word embeddings we use the English ukWaC corpus[10] [Baroni et al., 2009], which is made up of two billion words from paragraphs extracted from the web.

Knowledge-based

In this section we describe the three knowledge-based WSD models used in our empirical comparison:

[7] As already noted by Taghipour and Ng [2015a], supervised systems trained on only OMSTI obtain lower results than when trained along with SemCor, mainly due to OMSTI's lack of coverage in target word types.

[8] We used the implementation available at `https://github.com/iiacobac/ims_wsd_emb`

[9] `code.google.com/archive/p/word2vec/`

[10] `http://wacky.sslmit.unibo.it/doku.php?id=corpora`

- **Lesk** [Lesk, 1986] is a simple knowledge-based WSD algorithm that bases its calculations on the overlap between the definitions of a given sense and the context of the target word. For our experiments we replicated the extended version of the original algorithm in which definitions of related senses are also considered and the conventional term frequency-inverse document frequency [Jones, 1972, *tf-idf*] is used for word weighting [Banerjee and Pedersen, 2003, Lesk$_{\text{ext}}$]. Additionally, we included the enhanced version of Lesk in which word embeddings[11] are leveraged to compute the similarity between definitions and the target context [Basile et al., 2014, Lesk$_{\text{ext}}$+emb][12].

- **UKB** [Agirre and Soroa, 2009, Agirre et al., 2014] is a graph-based WSD system which makes use of random walks over a semantic network (WordNet graph in this case). UKB[13] applies the Personalized Page Rank algorithm [Haveliwala, 2002] initialized using the context of the target word. Unlike most WSD systems, UKB does not back-off to the WordNet first sense heuristic and it is self-contained (i.e., it does not make use of any external resources/corpora). We used both default configurations from UKB: using the full WordNet graph (UKB) and the full graph including disambiguated glosses as connections as well (UKB_gloss).

- **Babelfy** [Moro et al., 2014b], as described in chapter 2, is a graph-based disambiguation approach which exploits random walks to determine connections between synsets. Specifically, Babelfy[14] uses random walks with restart [Tong et al., 2006] over BabelNet [Navigli and Ponzetto, 2012]. Its algorithm is based on a densest subgraph heuristic for selecting high-coherence semantic interpretations of the input text. The best configuration of Babelfy takes into account not only the target sentence in which the target word occurs, but also the whole document.

As knowledge-based baseline we included the **WordNet first sense**. This baseline simply selects the candidate which is considered as first sense in WordNet 3.0. Even though the sense order was decided on the basis of semantically-tagged

[11] We used the same word embeddings for IMS+emb.

[12] We used the implementation from `https://github.com/pippokill/lesk-wsd-dsm`. In this implementation additional definitions from BabelNet are considered.

[13] We used the implementation available at `http://ixa2.si.ehu.es/ukb/`

[14] We used the Java API from `http://babelfy.org`

text, we considered it as knowledge-based in this experiment as this information is already available in WordNet. In fact, knowledge-based systems like Babelfy include this information in their pipeline. Despite its simplicity, this baseline has been shown to be hard to beat by automatic WSD systems [Navigli, 2009, Agirre et al., 2014].

	Tr. Corpus	**System**	**Senseval-2**	**Senseval-3**	**SemEval-07**	**SemEval-13**	**SemEval-15**
Supervised	**SemCor**	IMS	70.9	69.3	61.3	65.3	69.5
		IMS+emb	71.0	69.3	60.9	**67.3**	71.3
		IMS$_{\text{-s}}$+emb	**72.2**	**70.4**	**62.6**	65.9	71.5
		Context2Vec	71.8	69.1	61.3	65.6	**71.9**
		MFS	65.6	66.0	54.5	63.8	67.1
		Ceiling	*91.0*	*94.5*	*93.8*	*88.6*	*90.4*
	SemCor + OMSTI	IMS	72.8	69.2	60.0	65.0	69.3
		IMS+emb	70.8	68.9	58.5	66.3	69.7
		IMS$_{\text{-s}}$+emb	**73.3**	**69.6**	61.1	66.7	70.4
		Context2Vec	72.3	68.2	**61.5**	**67.2**	**71.7**
		MFS	66.5	60.4	52.3	62.6	64.2
		Ceiling	*91.5*	*94.9*	*94.7*	*89.6*	*91.1*
Knowledge	**-**	Lesk$_{\text{ext}}$	50.6	44.5	32.0	53.6	51.0
		Lesk$_{\text{ext}}$+emb	63.0	63.7	**56.7**	66.2	64.6
		UKB	56.0	51.7	39.0	53.6	55.2
		UKB_gloss	60.6	54.1	42.0	59.0	61.2
		Babelfy	**67.0**	63.5	51.6	**66.4**	**70.3**
		WN 1st sense	66.8	**66.2**	55.2	63.0	67.8

Table 4.2. F-Measure percentage of different models in five all-words WSD datasets.

4.3.2 Results

Table 4.2 shows the F-Measure performance of all comparison systems on the five all-words WSD datasets. Since not all test word instances are covered by the corresponding training corpora, supervised systems have a maximum F-Score (*ceiling* in the Table) they can achieve. Nevertheless, supervised systems consistently outperform knowledge-based systems across datasets, confirming the results of Pilehvar and Navigli [2014]. A simple linear classifier over conventional WSD features (i.e., IMS) proves to be robust across datasets, consistently outperforming the MFS baseline. The recent integration of word embeddings as an additional feature is beneficial, especially as a replacement of the feature based on the surface form of surrounding words (i.e., IMS$_{\text{-s}}$+emb). Moreover, recent advances on neural language models (in the case of Context2Vec a bi-directional LSTM) appear to be highly promising for the WSD task according to the results, as Context2Vec

	Nouns	Verbs	Adj.	Adv.	All
#Instances	4,300	1,652	955	346	7,253
Ambiguity	4.8	10.4	3.8	3.1	5.8

Table 4.3. Number of instances and ambiguity level of the concatenation of all five WSD datasets.

outperforms IMS in most datasets.

On the other hand, it is also interesting to note the performance inconsistencies of systems across datasets, as in all cases there is a large performance gap between the best and the worst performing dataset. As explained in Section 4.2.3, the ambiguity level may give a hint as to how difficult the corresponding dataset may be. In fact, WSD systems obtain relatively low results in SemEval-07, which is the most ambiguous dataset (see Table 4.1). However, this is the dataset in which supervised systems achieve a larger margin with respect to the MFS baseline, which suggests that, in general, the MFS heuristic does not perform accurately on highly ambiguous words.

4.3.3 Analysis

To complement the results from the previous section, we additionally carried out a detailed analysis about the global performance of each system and divided by PoS tag. To this end, we concatenated all five datasets into a single dataset. This resulted in a large evaluation dataset of 7,253 instances to disambiguate (see Table 4.3). Table 4.4 shows the F-Measure performance of all comparison systems on the concatenation of all five WSD evaluation datasets, divided by PoS tag. IMS$_{\text{-s}}$+emb trained on SemCor+OMSTI achieves the best overall results, slightly above Context2Vec trained on the same corpus. In what follows we describe some of the main findings extracted from our analysis.

Training corpus. In general, the results of supervised systems trained on SemCor only (manually-annotated) are lower than training simultaneously on both SemCor and OMSTI (automatically-annotated). This is a promising finding, which confirms the results of previous works [Iacobacci et al., 2016, Yuan et al., 2016] and encourages further research on developing reliable automatic or semi-automatic methods to obtain large amounts of sense-annotated corpora in order to overcome

	Tr. Corpus	System	Nouns	Verbs	Adjectives	Adverbs	All
Supervised	**SemCor**	IMS	70.4	56.1	75.6	82.9	68.4
		IMS+emb	71.8	55.4	**76.1**	82.7	69.1
		IMS$_{\text{-s}}$+emb	**71.9**	56.9	75.9	**84.7**	**69.6**
		Context2Vec	71.0	**57.6**	75.2	82.7	69.0
		MFS	67.6	49.6	73.1	80.5	64.8
		Ceiling	*89.6*	*95.1*	*91.5*	*96.4*	*91.5*
	SemCor + OMSTI	IMS	70.5	**56.9**	76.8	82.9	68.8
		IMS+emb	71.0	53.3	77.1	82.7	68.3
		IMS$_{\text{-s}}$+emb	**72.0**	56.5	76.6	**84.7**	**69.7**
		Context2Vec	71.7	55.8	**77.2**	82.7	69.4
		MFS	65.8	45.9	72.7	80.5	62.9
		Ceiling	*90.4*	*95.8*	*91.8*	*96.4*	*92.1*
Knowledge	**-**	Lesk$_{\text{ext}}$	54.1	27.9	54.6	60.3	48.7
		Lesk$_{\text{ext}}$+emb	**69.8**	**51.2**	51.7	80.6	63.7
		UKB	56.7	39.3	63.9	44.0	53.2
		UKB_gloss	62.1	38.3	66.8	66.2	57.5
		Babelfy	68.6	49.9	73.2	79.8	**65.5**
		WN 1$^{\text{st}}$ sense	67.6	50.3	**74.3**	**80.9**	65.2

Table 4.4. F-Measure percentage of different models on the concatenation of all five WSD datasets.

the knowledge-acquisition bottleneck. For instance, Context2Vec improves 0.4 points overall when adding the automatically sense-annotated OMSTI as part of the training corpus, suggesting that more data, even if not perfectly clean, may be beneficial for neural language models.

Knowledge-based vs. Supervised. One of the main conclusions that can be taken from the evaluation is that supervised systems clearly outperform knowledge-based models. This may be due to the fact that in many cases the main disambiguation clue is given by the immediate local context. This is particularly problematic for knowledge-based systems, as they take equally into account all the words within a sentence (or document in the case of Babelfy). For instance, in the following sentence, both UKB and Babelfy fail to predict the correct sense of *state*:

In sum, at both the federal and ***state*** *government levels at least part of the seemingly irrational behavior voters display in the voting booth may have an exceedingly rational explanation.*

In this sentence, *state* is annotated with its *administrative districts of a nation* sense in the gold standard. The main disambiguation clue seems to be given by its previous and immediate subsequent words (*federal* and *government*), which tend to co-occur with this particular sense. However, knowledge-based WSD systems like UKB or Babelfy give the same weight to all words in context, underrating the importance of this local disambiguation clue in the example. For instance, UKB disambiguates *state* with the sense defined as *the way something is with respect to its main attributes*, probably biased by words which are not immediately next to the target word within the sentence, e.g., *irrational*, *behaviour*, *rational* or *explanation*.

Low overall performance on verbs. As can be seen from Table 4.4, the F-Measure performance of all systems on verbs is in all cases below 58%. This can be explained by the high granularity of verbs in WordNet. For instance, the verb *keep* consists of 22 different meanings in WordNet 3.0, six of them denoting "possession and transfer of possession"[15]. In fact, the average ambiguity level of all verbs in this evaluation framework is 10.4 (see Table 4.3), considerably greater than the ambiguity on other PoS tags, e.g., 4.8 in nouns. Nonetheless, supervised systems manage to comfortably outperform the MFS baseline, which does not seem to be reliable for verbs given their high ambiguity.

Influence of preprocessing. As mentioned in Section 4.1, our evaluation framework provides a preprocessing of the corpora with Stanford CoreNLP. This ensures a fair comparison among all systems but may introduce some annotation inaccuracies, such as erroneous PoS tags. However, for English these errors are minimal[16]. For instance, the global error rate of the Stanford PoS tagger in all disambiguation instances is 3.9%, which were fixed as explained in Section 4.1.

Bias towards the Most Frequent Sense. After carrying out an analysis on the influence of MFS in WSD systems[17], we found that all supervised systems suffer a strong bias towards the MFS, with all IMS-based systems disambiguating over 75% of instances with their MFS. Context2Vec is slightly less affected by this bias, with

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

[16] Even if preprocessing plays a minimal role for English, it may be of higher importance for other languages, e.g., morphologically richer languages [Eger et al., 2016].

[17] See Postma et al. [2016] for an interesting discussion on the bias of current WSD systems towards the MFS.

71.5% (SemCor) and 74.7% (SemCor+OMSTI) of answers corresponding to the MFS. Interestingly, this MFS bias is also present in graph knowledge-based systems. In fact, Calvo and Gelbukh [2015] had already shown how the MFS correlates strongly with the number of connections in WordNet.

Knowledge-based systems. For knowledge-based systems the WN first sense baseline proves still to be extremely hard to beat. The only knowledge-based system that overall manages to beat this baseline is Babelfy, which, in fact, uses information about the first sense in its pipeline. Babelfy's default pipeline includes a confidence threshold in order to decide whether to disambiguate or back-off to the first sense. In total, Babelfy backs-off to WN first sense in 63% of all instances. Nonetheless, it is interesting to note the high performance of Babelfy and Lesk_{ext}+emb on noun instances (outperforming the first sense baseline by 1.0 and 2.2 points, respectively) in contrast to their relatively lower performance on verbs, adjectives[18] and adverbs. We believe that this is due to the nature of the lexical resource used by these two systems, i.e., BabelNet. BabelNet includes Wikipedia as one of its main sources of information. However, while Wikipedia provides a large amount of semantic connections and definitions for nouns, this it not the case for verbs, adjectives and adverbs, as they are not included in Wikipedia and their source of information mostly comes from WordNet only.

4.4 Conclusion

In this chapter we presented a unified evaluation framework for all-words WSD, addressing the first two objectives of this thesis. This framework is based on evaluation datasets taken from Senseval and SemEval competitions, as well as manually and automatically sense-annotated corpora. In this evaluation framework all datasets share a common format, sense inventory (i.e., WordNet 3.0) and pre-processing pipeline, which eases the task of researchers to evaluate their models and, more importantly, ensures a fair comparison among all systems. The whole evaluation framework[19], including guidelines for researchers to include their own sense-annotated datasets and a script to validate their conformity to the guidelines, is available at `http://lcl.uniroma1.it/wsdeval`. We used this framework

[18] The poor performance of Lesk_{ext}+emb on adjective instances is particularly noticeable.

[19] We have additionally set up a CodaLab competition based on this evaluation framework.

to perform an empirical comparison among a set of heterogeneous WSD systems, including both knowledge-based and supervised ones. Supervised systems based on neural networks achieve the most promising results.

Given our analysis, we foresee two potential research avenues focused on semi-supervised learning: (1) exploiting large amounts of unlabeled corpora for learning word embeddings or training neural sequence learning models, and (2) automatically constructing high-quality sense-annotated corpora to be used by supervised WSD systems.

Chapter 5

Seq2Sense: Neural Sequence Learning Models for Word Sense Disambiguation

As result of the analysis of the previous chapter, we focused our attention on neural models, studying several neural sequence systems trained directly from raw text to senses, without any engineered features, exploiting only word embeddings. In this chapter we describe Seq2Sense, neural sequence learning models directly tailored to WSD. The models are able to handle multiple target words at the same time and disambiguate them jointly, providing considerable contributions over the state of the art in WSD. First, we propose a novel approach to perform all-words WSD, showing that sequence-to-sequence learning can be leveraged to take the best of both worlds, and couple the flexibility of knowledge-based systems with the accuracy of supervised models. Second, we carry out an extensive experimental evaluation of our models with different configurations and training procedures and show that it leads to performances that are consistently in line with the state of the art across different benchmarks. Third, we show, for the first time in WSD, how to cope the knowledge acquisition bottleneck, describing how to evaluate the models in a cross-lingual settings, training on English and testing in other languages. Moreover, we also describe a specialized sequence-to-label architecture aimed at disambiguate one word at time, like the word expert paradigm.

The rest of this chapter is organized as follows. We first provide a description of the sequence learning models in Section 5.1. An augmentation version and a specialized variant is given in Sections 5.2 and 5.3, respectively. Section 5.4 and 5.5

provide the experimental setup and results comparing our models against the state of the art systems. We then describe in Section 5.5.2 our multilingual experiments and in Section 5.5.3 are presented and discussed some analysis and findings. Finally, we provide concluding remarks in Section 5.6.

5.1 Sequence Learning for Word Sense Disambiguation

In this section we define WSD in terms of a sequence learning problem. While in its classical formulation [Navigli, 2009] WSD is viewed as a classification problem for a given word w in context, with word senses of w being the class labels, here we consider a variable-length sequence of input symbols $\vec{x} = \langle x_1, ..., x_T \rangle$ and we aim at predicting a sequence of output symbols $\vec{y} = \langle y_1, ..., y_{T'} \rangle$.[1] Input symbols are word tokens drawn from a given vocabulary V.[2] Output symbols are either drawn from a pre-defined sense inventory S (if the corresponding input symbols are open-class content words, i.e., nouns, verbs, adjectives or adverbs), or from the same input vocabulary V (e.g., if the corresponding input symbols are function words, like prepositions or determiners). Hence, we can define a WSD model in terms of a function that maps sequences of symbols $x_i \in V$ into sequences of symbols $y_j \in O = S \cup V$.

Here all-words WSD is no longer broken down into a series of distinct and separate classification tasks (one per target word) but rather treated directly at the sequence level, with a single model handling all disambiguation decisions. In what follows, we describe three different models for accomplishing this: a traditional LSTM-based model (Section 5.1.1), a variant that incorporates an attention mechanism (Section 5.1.2), and an encoder-decoder architecture (Section 5.1.3).

5.1.1 Bidirectional LSTM Tagger

The most straightforward way of modeling WSD as formulated in Section 5.1 is that of considering a sequence labeling architecture that tags each symbol $x_i \in V$ in the input sequence with a label $y_j \in O$. Even though the formulation is rather

[1] In general $\vec{x}$ and $\vec{y}$ might have different lengths, e.g., if $\vec{x}$ contains a multi-word expression (*European Union*) which is mapped to a unique sense identifier ($\texttt{European Union}^1_n$).

[2] V generalizes traditional vocabularies used in WSD and includes both word lemmas and inflected forms.

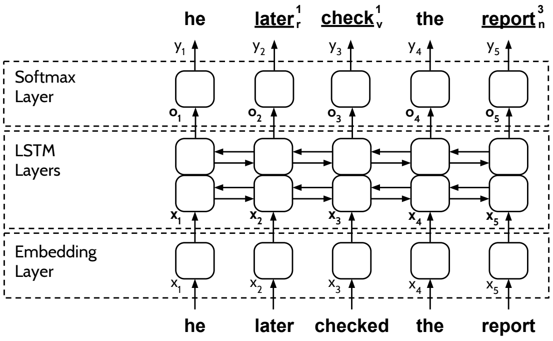


Figure 5.1. Bidirectional LSTM sequence labeling architecture for WSD (2 hidden layers). We use the notation of Navigli [2009] for word senses: w^i_p is the i-th sense of w with part of speech p.

general, previous contributions [Melamud et al., 2016, Kågebäck and Salomonsson, 2016] have already shown the effectiveness of recurrent neural networks for WSD. We follow the same line and employ a bidirectional LSTM architecture: in fact, important clues for disambiguating a target word could be located anywhere in the context (not necessarily before the target) and for a model to be effective it is crucial that it exploits information from the whole input sequence at every time step.

Architecture. A sketch of our bidirectional LSTM tagger is shown in Figure 5.1. It consists of:

- An embedding layer that converts each word $x_i \in \vec{x}$ into a real-valued d-dimensional vector $\mathbf{x}_i$ via the embedding matrix $\mathbf{W} \in \mathbb{R}^{d \times |V|}$;

- One or more stacked layers of bidirectional LSTM [Graves and Schmidhuber, 2005]. The hidden state vectors $\mathbf{h}_i$ and output vectors $\mathbf{o}_i$ at the i^{th} time step are then obtained as the concatenations of the forward and backward pass vectors $\overrightarrow{\mathbf{h}}_i$, $\overrightarrow{\mathbf{o}}_i$ and $\overleftarrow{\mathbf{h}}_i$, $\overleftarrow{\mathbf{o}}_i$;

- A fully-connected layer with softmax activation that turns the output vector $\mathbf{o}_i$ at the i^{th} time step into a probability distribution over the output vocabulary O.

Training. The tagger is trained on a dataset of N labeled sequences $\{(\vec{x}_k, \vec{y}_k)\}_{k=1}^N$ directly obtained from the sentences of a sense-annotated corpus, where each $\vec{x}_k$ is a sequence of word tokens, and each $\vec{y}_k$ is a sequence containing both word tokens and sense labels. Ideally $\vec{y}_k$ is a copy of $\vec{x}_k$ where each content word is sense-tagged.

This is, however, not the case in many real-world datasets, where only a subset of the content words is annotated; hence the architecture is designed to deal with both fully and partially annotated sentences. Apart from sentence splitting and tokenization, no preprocessing is required on the training data.

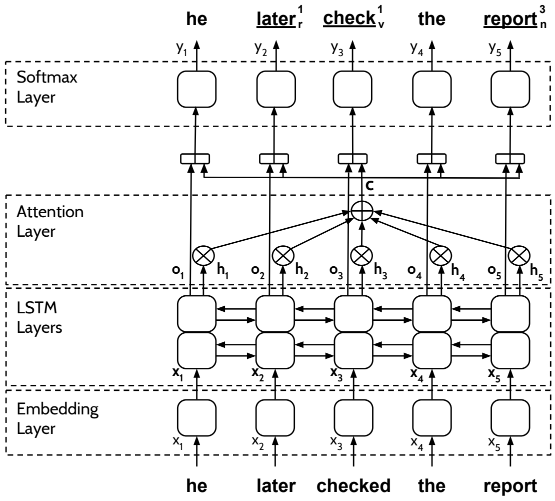


Figure 5.2. Attentive bidirectional LSTM sequence labeling architecture for WSD (2 hidden layers).

5.1.2 Attentive Bidirectional LSTM Tagger

The bidirectional LSTM tagger of Section 5.1.1 exploits information from the whole input sequence $\vec{x}$, which is encoded in the hidden state $\mathbf{h}_i$. However, certain elements of $\vec{x}$ might be more discriminative than others in predicting the output label at a given time step (e.g., the syntactic subject and object when predicting the sense label of a verb).

We model this hunch by introducing an attention mechanism, already proven to be effective in other NLP tasks [Bahdanau et al., 2015, Vinyals et al., 2015], into the sequence labeling architecture of Section 5.1.1. The resulting *attentive* bidirectional LSTM tagger augments the original architecture with an attention layer, where a context vector $\mathbf{c}$ is computed from all the hidden states $\mathbf{h}_1, ..., \mathbf{h}_T$ of the bidirectional LSTM. The attentive tagger first reads the entire input sequence $\vec{x}$ to construct $\mathbf{c}$, and then exploits $\mathbf{c}$ to predict the output label y_j at each time step, by concatenating it with the output vector $\mathbf{o}_j$ of the bidirectional LSTM (Figure 5.2).

We follow previous work [Vinyals et al., 2015, Zhou et al., 2016] and compute **c** as the weighted sum of the hidden state vectors $\mathbf{h}_1, ..., \mathbf{h}_T$. Formally, let $H \in \mathbb{R}^{n \times T}$ be the matrix of hidden state vectors $[\mathbf{h}_1, ..., \mathbf{h}_T]$, where n is the hidden state dimension and T is the input sequence length (cf. Section 5.1). **c** is obtained as follows:

$$\begin{aligned} \mathbf{u} &= \omega^T \tanh(H) \\ \mathbf{a} &= softmax(\mathbf{u}) \\ \mathbf{c} &= H\mathbf{a}^T \end{aligned} \tag{5.1}$$

where $\omega \in \mathbb{R}^n$ is a parameter vector, and $\mathbf{a} \in \mathbb{R}^T$ is the vector of normalized attention weights.

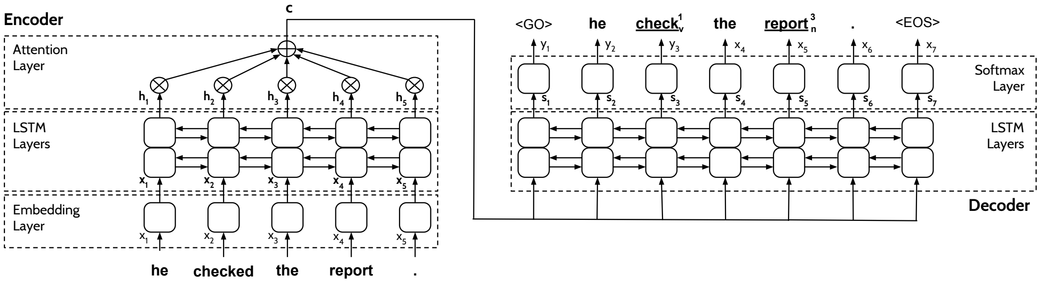


Figure 5.3. Encoder-decoder architecture for sequence-to-sequence WSD, with 2 bidirectional LSTM layers and an attention layer.

5.1.3 Sequence-to-Sequence Model

The attentive tagger of Section 5.1.2 performs a two-pass procedure by first reading the input sequence $\vec{x}$ to construct the context vector **c**, and then predicting an output label y_j for each element in $\vec{x}$. In this respect, the attentive architecture can effectively be viewed as an encoder for $\vec{x}$. A further generalization of this model would then be a complete encoder-decoder architecture [Sutskever et al., 2014] where WSD is treated as a sequence-to-sequence mapping (*sequence-to-sequence WSD*), i.e., as the "translation" of word sequences into sequences of potentially sense-tagged tokens.

In the sequence-to-sequence framework, a variable-length sequence of input symbols $\vec{x}$ is represented as a sequence of vectors $\vec{\mathbf{x}} = \langle \mathbf{x}_1, ..., \mathbf{x}_T \rangle$ by converting each symbol $x_i \in \vec{x}$ into a real-valued vector $\mathbf{x}_i$ via an embedding layer, and then fed to an encoder, which generates a fixed-dimensional vector representation of the sequence. Traditionally, the encoder function is a Recurrent Neural Network (RNN) such that:

$$\begin{aligned} \mathbf{h}_t &= f(\mathbf{h}_{t-1}, \mathbf{x}_t) \\ \mathbf{c} &= q(\{\mathbf{h}_1, ..., \mathbf{h}_T\}) \end{aligned} \tag{5.2}$$

where $\mathbf{h}_t \in \mathbb{R}^n$ is the n-dimensional hidden state vector at time t, $\mathbf{c} \in \mathbb{R}^n$ is a vector generated from the whole sequence of input states, and f and q are non-linear functions.[3] A decoder is then trained to predict the next output symbol y_t given the encoded input vector $\mathbf{c}$ and all the previously predicted output symbols $\langle y_1, ..., y_{t-1} \rangle$. More formally, the decoder defines a probability over the output sequence $\vec{y} = \langle y_1, ..., y_{T'} \rangle$ by decomposing the joint probability into ordered conditionals:

$$p(\vec{y} \,|\, \vec{x}) = \prod_{t=1}^{T'} p(y_t \,|\, \mathbf{c}, \langle y_1, ..., y_{t-1} \rangle) \tag{5.3}$$

Typically a decoder RNN defines the hidden state at time t as $\mathbf{s}_t = g(\mathbf{s}_{t-1}, \{\mathbf{c},\ y_{t-1}\})$ and then feeds $\mathbf{s}_t$ to a softmax layer in order to obtain a conditional probability over output symbols.

In the context of WSD framed as a sequence learning problem, a sequence-to-sequence model takes as input a training set of labeled sequences (cf. Section 5.1.1) and learns to replicate an input sequence $\vec{x}$ while replacing each content word with its most suitable word sense from S. In other words, sequence-to-sequence WSD can be viewed as the combination of two sub-tasks:

- A *memorization* task, where the model learns to replicate the input sequence token by token at decoding time;
- The actual *disambiguation* task where the model learns to replace content words across the input sequence with their most suitable senses from the sense inventory S.

In the latter stage, multi-word expressions (such as nominal entity mentions or phrasal verbs) are replaced by their sense identifiers, hence yielding an output sequence that might have a different length than $\vec{x}$.

[3] For instance, Sutskever et al. [2014] used an LSTM as f, and $q(\{\mathbf{h}_1, ..., \mathbf{h}_T\}) = \mathbf{h}_T$.

Architecture. The encoder-decoder architecture generalizes over both the models in Sections 5.1.1 and 5.1.2. In particular, we include one or more bidirectional LSTM layers at the core of both the encoder and the decoder modules. The encoder utilizes an embedding layer (cf. Section 5.1.1) to convert input symbols into embedded representations, feeds it to the bidirectional LSTM layer, and then constructs the context vector $\mathbf{c}$, either by simply letting $\mathbf{c} = \mathbf{h}_T$ (i.e., the hidden state of the bidirectional LSTM layer after reading the whole input sequence), or by computing the weighted sum described in Section 5.1.2 (if an attention mechanism is employed). In either case, the context vector $\mathbf{c}$ is passed over to the decoder, which generates the output symbols sequentially based on $\mathbf{c}$ and the current hidden state $\mathbf{s}_t$, using one or more bidirectional LSTM layers as in the encoder module. Instead of feeding $\mathbf{c}$ to the decoder only at the first time step [Sutskever et al., 2014, Vinyals and Le, 2015], we condition each output symbol y_t on $\mathbf{c}$, allowing the decoder to peek into the input at every step, as in Cho et al. [2014]. Finally, a fully-connected layer with softmax activation converts the current output vector of the last LSTM layer into a probability distribution over the output vocabulary O. The complete encoder-decoder architecture (including the attention mechanism) is shown in Figure 5.3.

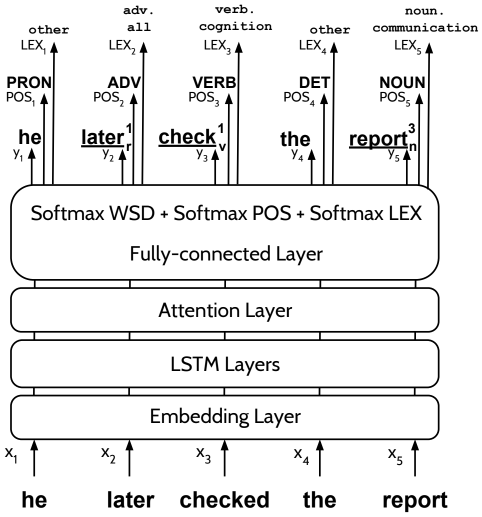


Figure 5.4. Multitask augmentation (with both POS and LEX as auxiliary tasks) for the attentive bidirectional LSTM tagger of Section 5.1.2.

5.2 Multitask Learning with Multiple Auxiliary Losses

Several recent contributions [Søgaard and Goldberg, 2016, Bjerva et al., 2016, Plank et al., 2016, Luong et al., 2016] have shown the effectiveness of *multitask learning* [Caruana, 1997, MTL] in a sequence learning scenario. In MTL the idea is that of improving generalization performance by leveraging training signals contained in related tasks, in order to exploit their commonalities and differences. MTL is typically carried out by training a single architecture using multiple loss functions and a shared representation, with the underlying intention of improving a main task by incorporating joint learning of one or more related auxiliary tasks. From a practical point of view, MTL works by including one task-specific output layer per additional task, usually at the outermost level of the architecture, while keeping the remaining hidden layers common across all tasks.

In line with previous approaches, and guided by the intuition that WSD is strongly linked to other NLP tasks at various levels, we also design and study experimentally a multitask augmentation of the models described in Section 5.1. In particular, we consider two auxiliary tasks:

- **Part-of-speech (POS) tagging**, a standard auxiliary task extensively studied in previous work [Søgaard and Goldberg, 2016, Plank et al., 2016]. Predicting the part-of-speech tag for a given token can also be informative for word senses, and help in dealing with cross-POS lexical ambiguities (e.g., *book a flight* vs. *reading a good book*);

- **Coarse-grained semantic labels (LEX)** based on the WordNet [Miller, 1995] lexicographer files,[4] i.e., 45 coarse-grained semantic categories manually associated with all the synsets in WordNet on the basis of both syntactic and logical groupings (e.g., *noun.location*, or *verb.motion*). These very coarse semantic labels, recently employed in a multitask setting by Martínez Alonso and Plank [2017], group together related senses and help the model to generalize, especially over senses less covered at training time.

We follow previous work [Plank et al., 2016, Martínez Alonso and Plank, 2017] and define an auxiliary loss function for each additional task. The overall loss is then computed by summing the main loss (i.e., the one associated with word sense labels) and all the auxiliary losses taken into account.

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

As regards the architecture, we consider both the models described in Sections 5.1.2 and 5.1.3 and modify them by adding two softmax layers in addition to the one in the original architecture. Figure 4 illustrates this for the attentive tagger of Section 5.1.2, considering both POS and LEX as auxiliary tasks. At the j^{th} time step the model predicts a sense label y_j together with a part-of-speech tag POS$_j$ and a coarse semantic label LEX$_j$.[5]

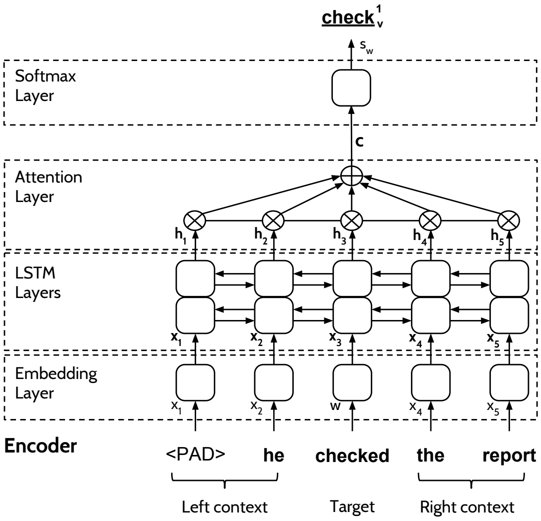


Figure 5.5. SEQ2SENSE specialized architecture for sequence-to-label WSD.

5.3 Sequence-to-Label Word Sense Disambiguation

The sequence-to-sequence model described throughout Section 5.1 is explicitly designed for joint all-words disambiguation of a given text. In many WSD settings, however, the focus of disambiguation is a specific word w within the input text, and the remaining words are only intended as context for w. We therefore revise the general structure of SEQ2SENSE, and design a variant of the disambiguation model that is specialized for the WSD setting just described, in which a single disambiguation target w is provided within an input context.

With this revised version of SEQ2SENSE, WSD is formulated in terms of learning a mapping from sequences of words $\vec{x}_i$ to individual sense labels s_w (*sequence-to-label WSD*), instead of entire sequences of word tokens and sense labels. These labels s_w provide, for each input sequence, the most suitable word sense of the

[5] We use a dummy LEX label (`other`) for punctuation and function words.

target word w according to the sense inventory S. The resulting SEQ2SENSE model now deals with a simplified learning problem consisting only of the straightforward disambiguation of w.

Formulation. Formally, in this specialized sequence-to-label framework, supervised WSD is framed as the task of learning a mapping from fixed-length sequences $\vec{x}$ of symbols in V to output sense labels s_w in S, where w is the target word. Input sequences are structured as follows:

$$\begin{aligned} \vec{x}_i &= \langle x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_T \rangle \\ &= \langle \overrightarrow{x}_w,\, w,\, \overleftarrow{x}_w \rangle, \qquad x_i = w \end{aligned} \tag{5.4}$$

where $\overrightarrow{x}_w = \langle x_1, ..., x_{i-1} \rangle$ is the *left-context sequence*, $\overleftarrow{x}_w = \langle x_{i+1}, ..., x_T \rangle$ is the *right-context sequence*. The corresponding output label s_w is the intended sense of w in the context provided by $\vec{x}$. In this setting, the encoder function is the same as Equation 5.2, while the factorized decoder probability of Equation 5.3 reduces to $p(s_w \,|\, \vec{x}) = p(s_w \,|\, \mathbf{c})$.

Architecture. The revised architecture is shown in Figure 5.5. With respect to the model described in Section 5.1.3, the encoder module is left unchanged while the decoder module is entirely replaced by a single fully-connected softmax layer that turns the input representation vector **c** into a probability distribution over the sense inventory S. This distribution is used to predict the most suitable sense for the target content word w. Compared to the original model, this specialized architecture computes a single softmax once the encoding phase is complete, and directly over the sense inventory S (instead of O).

The architecture in Figure 5.5 is similar to the word expert proposed by Kågebäck and Salomonsson [2016], with two crucial differences: first, the bidirectional LSTM layers perform a full forward and backward pass over the whole sequence, and the final output is weighted via an attention mechanism (cf. Section 5.1.3); second, this revised sequence-to-label variant of SEQ2SENSE remains a single model capable of outputting disambiguation decisions for any target content word in V.

This specialized version is trained end-to-end on fixed-length sequences. We obtain a training instance for each sense-annotated word w across the corpus by considering sequences of length T (fixed to 31) centered on w (15 words as left and right context).

5.4 Experimental Setup

In this section we detail the setup of our experimental evaluation. We first describe the training corpus and all the standard benchmarks for all-words WSD; we then report technical details on the architecture and on the training process for all the models described throughout Section 5.1 and their multitask augmentations (Section 5.2).

Evaluation Benchmarks. We evaluated our models on the English all-words WSD task, considering both the fine-grained and coarse-grained benchmarks (Section 5.5.1). As regards fine-grained WSD, we relied on the evaluation framework of chapter 4, which includes five standardized test sets from the Senseval/SemEval series: Senseval-2 [Edmonds and Cotton, 2001, **SE2**], Senseval-3 [Snyder and Palmer, 2004, **SE3**], SemEval-2007 [Pradhan et al., 2007, **SE07**], SemEval-2013 [Navigli et al., 2013, **SE13**] and SemEval-2015 [Moro and Navigli, 2015, **SE15**]. Due to the lack of a reasonably large development set for our setup, we considered the smallest among these test sets, i.e., **SE07**, as development set and excluded it from the evaluation of Section 5.5.1. As for coarse-grained WSD, we used the SemEval-2007 task 7 test set [Navigli et al., 2007], which is not included in the standardized framework, and mapped the original sense inventory from WordNet 2.1 to WordNet 3.0.[6] Finally, we carried out an experiment on multilingual WSD using the Italian, German, French and Spanish data of **SE13**. For these benchmarks we relied on BabelNet [Navigli and Ponzetto, 2012][7] as unified sense inventory.

At testing time, given a target word w, our models used the probability distribution over the output vocabulary, computed by the softmax layer at the corresponding time step, to rank the candidate senses of w; we then simply selected the top ranking candidate as output of the model.

Architecture Details. To set a level playing field with comparison systems on English all-words WSD, we followed chapter 4 and, for all our models, we used

[6] We utilized the original sense-key mappings available at `http://wordnetcode.princeton.edu/3.0` for nouns and verbs, and the automatic mappings by Daude et al. [2003] for the remaining parts of speech (not available in the original mappings).

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

a layer of word embeddings pre-trained[8] on the English ukWaC corpus [Baroni et al., 2009] as initialization, and kept them fixed during the training process. For all architectures we then employed 2 layers of bidirectional LSTM with 2048 hidden units (1024 units per direction).

As regards multilingual all-words WSD (Section 5.5.2), we experimented, instead, with two different configurations of the embedding layer: the pre-trained bilingual embeddings by Mrkšić et al. [2017] for all the language pairs of interest (EN-IT, EN-FR, EN-DE, and EN-ES), and the pre-trained multilingual 512-dimensional embeddings for 12 languages by Ammar et al. [2016].

Training. We used SemCor 3.0 [Miller et al., 1993] as training corpus for all our experiments. Widely known and utilized in the WSD literature, SemCor is one of the largest corpora annotated manually with word senses from the sense inventory of WordNet [Miller, 1995] for all open-class parts of speech. We used the standardized version of SemCor as provided in chapter 4 which also includes coarse-grained PoS tags from the universal tagset. All models were trained for a fixed number of epochs $E = 40$ using Adadelta [Zeiler, 2012] with learning rate 1.0 and batch size 32. After each epoch we evaluated our models on the development set, and then compared the best iterations (E^*) on the development set with the reported state of the art in each benchmark.

5.5 Experimental Results

Throughout this section we identify the models based on the LSTM tagger (Sections 5.1.1-5.1.2) by the label **BLSTM**, the sequence-to-sequence models (Section 5.1.3) by the label **Seq2Seq**, and the specialized variant (Section 5.3) by the label **Seq2Lab**.

5.5.1 English All-words WSD

Table 5.1 shows the performance of our models on the standardized benchmarks for all-words fine-grained WSD. We report the F1-score on each individual test set, as well as the F1-score obtained on the concatenation of all four test sets, divided by

[8] We followed Iacobacci et al. [2016] and used the Word2Vec [Mikolov et al., 2013a] skip-gram model with 400 dimensions, 10 negative samples and a window size of 10.

	Dev	Test Datasets				Concatenation of All Test Datasets				
	SE07	**SE2**	**SE3**	**SE13**	**SE15**	**Nouns**	**Verbs**	**Adj.**	**Adv.**	**All**
BLSTM	61.8	71.4	68.8	65.6	69.2	70.2	56.3	75.2	**84.4**	68.9
BLSTM + att.	62.4	71.4	**70.2**	66.4	70.8	71.0	**58.4**	75.2	83.5	69.7
BLSTM + att. + LEX	63.7	**72.0**	69.4	66.4	**72.4**	**71.6**	57.1	**75.6**	83.2	**69.9**
BLSTM + att. + LEX + POS	**64.8**	**72.0**	69.1	**66.9**	71.5	71.5	57.5	75.0	83.8	**69.9**
Seq2Seq	60.9	68.5	67.9	65.3	67.0	68.7	54.5	74.0	81.2	67.3
Seq2Seq + att.	62.9	69.9	69.6	65.6	67.7	69.5	57.2	74.5	81.8	68.4
Seq2Seq + att. + LEX	64.6	70.6	67.8	66.5	68.7	70.4	55.7	73.3	82.9	68.5
Seq2Seq + att. + LEX + POS	63.1	70.1	68.5	66.5	69.2	70.1	55.2	75.1	84.4	68.6
Seq2Lab	61.1	71.5	68.6	65.8	70.3	70.7	57.1	74.9	82.1	69.1
IMS	61.3	70.9	69.3	65.3	69.5	70.5	55.8	75.6	82.9	68.9
IMS+emb	**62.6**	**72.2**	**70.4**	65.9	71.5	**71.9**	56.6	**75.9**	**84.7**	**70.1**
Context2Vec	61.3	71.8	69.1	65.6	**71.9**	71.2	**57.4**	75.2	82.7	69.6
Lesk$_{ext}$+emb	⋆56.7	63.0	63.7	66.2	64.6	70.0	51.1	51.7	80.6	64.2
UKB$_{gloss}$ w2w	42.9	63.5	55.4	⋆62.9	63.3	64.9	41.4	69.5	69.7	61.1
Babelfy	51.6	⋆67.0	63.5	**66.4**	70.3	68.9	50.7	73.2	79.8	66.4
MFS	54.5	65.6	⋆66.0	63.8	⋆67.1	67.7	49.8	73.1	80.5	65.5

Table 5.1. F-scores (%) for English all-words fine-grained WSD on the test sets (including the development set **SE07**). The first system with a statistically significant difference from our best models is marked with ⋆ (unpaired t-test, $p < 0.05$).

part-of-speech tag.

We compared against the best supervised and knowledge-based systems evaluated on the same framework. As supervised systems, we considered **Context2Vec** [Melamud et al., 2016] and It Makes Sense [Zhong and Ng, 2010, **IMS**], both the original implementation and the best configuration reported by [Iacobacci et al., 2016, **IMS+emb**], which also integrates word embeddings using exponential decay.[9] All these supervised systems were trained on the standardized version of SemCor. As knowledge-based systems we considered the embeddings-enhanced version of Lesk by **Lesk$_{ext}$+emb** [Basile et al., 2014], UKB [Agirre et al., 2014] (**UKB$_{gloss}$ w2w**) [10] , and **Babelfy** [Moro et al., 2014b]. All these systems relied on the Most Frequent Sense (**MFS**) baseline as back-off strategy.[11] Overall, **BLSTM**, **Seq2Seq** and **Seq2Lab** achieved results that are either state-of-the-art or statistically equivalent (unpaired t-test, $p < 0.05$) to the best supervised system in each benchmark, performing on par with word experts tuned over explicitly engineered

[9] We are not including Yuan et al. [2016], as their models are not available and not replicable on the standardized test sets, being based on proprietary data.

[10] We report the best configuration of UKB ($w2w$) which uses the full WordNet graph and the disambiguated glosses of WordNet as connections.

[11] Since each system always outputs an answer, F-score equals both precision and recall, and statistical significance can be expressed with respect to any of these measures.

SemEval-2007 task 7			
BLSTM + att. + LEX	83.0	IMS	81.9
BLSTM + att. + LEX + POS	**83.1**	Chen et al. [2014]	82.6
Seq2Seq + att. + LEX	82.3	Yuan et al. [2016]	**82.8**
Seq2Seq + att. + LEX + POS	81.6	UKB w2w	80.1
Seq2Lab	82.0	MFS	78.9

Table 5.2. F-scores (%) for coarse-grained WSD.

features [Iacobacci et al., 2016]. Interestingly enough, **BLSTM** models tended consistently to outperform their **Seq2Seq** and **Seq2Lab** counterparts, suggesting that an encoder-decoder architecture, might be suboptimal for WSD, and that the specialized variant might have a too simplified architecture. Furthermore, introducing LEX (cf. Section 5.2) as auxiliary task was generally helpful; on the other hand, POS did not seem to help, corroborating previous findings [Martínez Alonso and Plank, 2017, Bingel and Søgaard, 2017].

The overall performance by part of speech was consistent with the above analysis, showing that our models outperformed all knowledge-based systems, while obtaining results that are superior or equivalent to the best supervised models. It is worth noting that RNN-based architectures outperformed classical supervised approaches [Zhong and Ng, 2010, Iacobacci et al., 2016] when dealing with verbs, which are shown to be highly ambiguous (see chapter 4).

The performance on coarse-grained WSD followed the same trend (Table 5.2). **BLSTM**, **Seq2Seq** and **Seq2Lab** outperformed UKB [Agirre et al., 2014] and IMS trained on SemCor [Taghipour and Ng, 2015a], as well as recent supervised approaches based on distributional semantics and neural architectures [Chen et al., 2014, Yuan et al., 2016].

	SemEval-2013 task 12			
	IT	**FR**	**DE**	**ES**
BLSTM (bilingual)	61.6	55.2	**69.2**	65.0
BLSTM (multilingual)	62.0	55.5	**69.2**	66.4
UMCC-DLSI	**65.8**	**60.5**	62.1	**71.0**
DAEBAK!	61.3	53.8	59.1	60.0
MFS	57.5	45.3	67.4	64.5

Table 5.3. F-scores (%) for multilingual WSD.

5.5.2 Multilingual All-words WSD

All the neural architectures described in this chapter can be readily adapted to work with different languages without adding sense-annotated data in the target language. In fact, as long as the first layer (cf. Figures 5.1-5.3) is equipped with *bilingual* or *multilingual* embeddings where word vectors in the training and target language are defined in the same space, the training process can be left unchanged, even if based only on English data. The underlying assumption is that words that are translations of each other (e.g., *house* in English and *casa* in Italian) are mapped to word embeddings that are as close as possible in the vector space.

In order to assess this, we considered one of our best models (**BLSTM+att.+LEX**) and replaced the monolingual embeddings with bilingual and multilingual embeddings (as specified in Section 5.4), leaving the rest of the architecture unchanged. We then trained these architectures on the same English training data, and ran the resulting models on the multilingual benchmarks of SemEval-2013 for Italian, French, German and Spanish. While doing this, we exploited BabelNet's inter-resource mappings to convert WordNet sense labels (used at training time) into BabelNet synsets compliant with the sense inventory of the task.

F-score figures (Table 5.3) show that bilingual and multilingual models, despite being trained only on English data, consistently outperformed the MFS baseline and achieved results that are competitive with the best participating systems in the task. We also note that the overall F-score performance did not change substantially (and slightly improved) when moving from bilingual to multilingual models, despite the increase in the number of target languages treated simultaneously.

5.5.3 Discussion and Error Analysis

All the neural models evaluated in Section 5.5.1 utilized the MFS back-off strategy for instances unseen at training time, which amounted to 9.4% overall for fine-grained WSD and 10.5% for coarse-grained WSD. Back-off strategy aside, 85% of the times the top candidate sense for a target instance lay within the 10 most probable entries in the probability distribution over O computed by the softmax layer.[12] In fact, our sequence models learned, on the one hand, to associate a target word with its candidate senses (something word experts are not required to learn, as they only deal with a single word type at a time); on the other, they tended to

[12] We refer here to the same model considered in Section 5.5.2 (i.e., **BLSTM+att.+LEX**).

generate softmax distributions reflecting the semantics of the surrounding context. For example, in the sentence:

(a) The two *justices* have been attending federalist society events for years,

our model correctly disambiguated *justices* with the WordNet sense $\texttt{justice}_n^3$ (public official) rather than $\texttt{justice}_n^1$ (the quality of being just), and the corresponding softmax distribution was heavily biased towards words and senses related to persons or groups (*commissioners*, *defendants*, *jury*, *cabinet*, *directors*). On the other hand, in the sentence:

(b) Xavi Hernandez, the player of Barcelona, has 106 *matches*,

the same model disambiguated *matches* with the wrong WordNet sense $\texttt{match}_n^1$ (tool for starting a fire). This suggests that the signal carried by discriminative words like *player* vanishes rather quickly. In order to enforce global coherence further, recent contributions have proposed more sophisticated models where recurrent architectures are combined with Conditional Random Fields [Huang et al., 2015, Ma and Hovy, 2016]. Finally, a number of errors were connected to shorter sentences with limited context for disambiguation: in fact, we noted that the average precision of our model, without MFS back-off, increased by 6.2% (from 74.6% to 80.8%) on sentences with more than 20 word tokens.

5.6 Conclusion

In this chapter we adopted a new perspective on supervised WSD, so far typically viewed as a classification problem at the word level, and framed it using neural sequence learning. To this aim we defined, analyzed and compared experimentally different end-to-end models of varying complexities, including augmentations based on an attention mechanism and multitask learning.

Unlike previous supervised approaches, where a dedicated model needs to be trained for every content word and each disambiguation target is treated in isolation, sequence learning approaches learn a single model in one pass from the training data, and then disambiguate jointly all target words within an input text. The resulting models consistently achieved state-of-the-art (or statistically equivalent) figures in all benchmarks for all-words WSD, both fine-grained and coarse-grained, effectively demonstrating that we can overcome the so far undisputed and long-standing word-expert assumption of supervised WSD, while retaining the accuracy of supervised word experts.

Furthermore, these models are sufficiently flexible to allow them, for the first time in WSD, to be readily adapted to languages different from the one used at training time, and still achieve competitive results (as shown in Section 5.5.2). This crucial feature could potentially pave the way for cross-lingual supervised WSD, and overcome the shortage of sense-annotated data in multiple languages that, to date, has prevented the development of supervised models for multiple languages.

Chapter 6

Automatic Construction and Evaluation of Sense-Tagged Corpora

We now address the knowledge acquisition bottleneck, i.e., the difficulty of obtaining knowledge in a computer-usable form [Buchanan and Wilkins, 1993], another objective of this thesis. From the previous chapter we showed how flexible neural models are, however, the WSD field still lacks of the availability of word-sense annotated corpora on a large scale. Gathering sense annotated corpora is a very hard task, talking about of millions of annotations be can be really demanding and time consuming. This is especially the case when such encoding requires both *lexicographic* (word senses) and *encyclopedic* knowledge (named entities) to be addressed [Schubert, 2006].Even though Amazon Mechanical Turk [Snow et al., 2008] or collaborative resource [Mihalcea, 2007] as Wikipedia can be used in order to obtain annotations, producing manually annotated corpus require an enormous effort. Recently, most works aim towards an automatic acquisition of large scale annotations [Zhong and Ng, 2009, Singh et al., 2012, Venhuizen et al., 2013, Gabrilovich et al., 2013, Moro et al., 2014a, Vannella et al., 2014, Jurgens and Navigli, 2014, Pasini and Navigli, 2017]. However, all these works present in general different problems: either they are still on small scale respect to the sense inventory, or contain only lexicographic annotations without considering named entities or vice-versa, or they are not ready available to the community. Moreover, it is even worse when we want to scale up covering more languages.

In this chapter, we present three ways to automatically annotate raw text on large scale and in multiple languages.

The remainder of this chapter is organized as follows: in Section 6.1, we describe how to use a multilingual knowledge-based system, i.e., Babelfy, at best to get semantic annotations from a large corpus. In Section 6.2 we present a method to construct and exploit a multilingual corpus in order to extend sense annotations also to multiple languages. Exploiting the wide coverage of BabelNet and parallel corpora providing enriched context, we are able to refine the quality of the annotations gathered using semantic similarity distribution. In Section 6.3, we show how to leverage a semi-structured resource to get automatically annotations without tuning any off-the-shelf system nor using any manual effort. Our extensive evaluations, beside providing the quality of the annotations, sets important performance baselines for multiple tasks and datasets. Finally, we provide concluding remarks in Section 6.4.

6.1 Annotating corpora with Babelfy

The most straightforward method to obtain sense annotations on large scale, is by using a knowledge-based system to annotate a big corpus. Indeed, in our settings we used the latest version of Babelfy[1], i.e., version 1.0, on Wikipedia. This release features many parameters among which adding pre-annotated fragments of text to help the disambiguation phase and to enable or disable the most common sense (MCS) backoff strategy that returns the most common sense for the text fragment when the system does not have enough information to select a meaning. Therefore we exploit the links of Wikipedia which are contained in BabelNet as pre-annotated fragments of text. By exploiting the Babelfy disambiguation system we leverage these hand-made connections to improve the quality of our automatic annotation. Each Wikipedia page, together with its internal links, corresponds to a Babel synset. Thus providing that information (i.e., the Babel synset) as disambiguation context for the text associated with the link in the page helps the Babelfy algorithm exclude less relevant candidates.

6.1.1 Statistics and Evaluation

In this section we present the statistics of our automatically annotated dataset. We used a sample of 500K articles of English Wikipedia and over 450K articles of Italian Wikipedia POS tagged with the Stanford POS Tagger [Manning et al., 2014]

[1] http://babelfy.org

	English		Italian	
# Articles		500,000		474,887
# Content Words	209,066,032		133,022,968	
# Non-Content Words	292,796,219		177,786,434	
# Words		501,862,251		310,809,402

Table 6.1. Statistics of the Wikipedia sample.

(for Italian we trained a model using the dataset from the Universal Dependency Treebank Project[2]). The corpora contain respectively 501M and 310M words (see Table 6.1), among which in both cases 42% are content words (i.e., words PoS tagged as noun, adjective, adverb or verb). In Table 6.2 and 6.3, we show the total number of our automatic annotations divided between concepts and named entities with and without the most common sense backoff strategy. As expected we have more annotations with the MCS, while without it we annotated 31% and 21% of the content words, respectively in English and Italian.

	English		Italian	
# Adjective Word Senses	14,662,188		5,921,520	
# Adverb Word Senses	3,402,554		2,604,358	
# Noun Word Senses	55,597,241		31,003,356	
# Verb Word Senses	26,072,320		11,942,285	
# Word Senses		99,734,303		51,471,519
# Named Entities		14,162,561		5,503,556
# Total Number of annotations		113,896,864		56,975,075

Table 6.2. Statistics of our automatic annotation of the Wikipedia corpus with MCS.

We performed an evaluation over a restricted sample of annotations to estimate the performance of the system using the accuracy measure, which is defined as the number of correct meanings/entities over the whole number of manually annotated mentions. We manually evaluated a random sample of 200 concepts and 200 named entities for both languages. We obtain an estimated accuracy of 77.8% for word senses and 63.2% for named entities for English, and 78.6% and 66% respectively for Italian.

[2] https://code.google.com/p/uni-dep-tb/

	English		Italian	
# Adjective Word Senses	7,816,765		2,848,886	
# Adverb Word Senses	2,450,533		1,385,650	
# Noun Word Senses	32,398,013		14,313,556	
# Verb Word Senses	8,683,852		3,302,068	
# Word Senses		51,349,163		21,850,160
# Named Entities		14,162,220		5,469,766
# Total Number of annotations		65,511,383		27,319,926

Table 6.3. Statistics of our automatic annotation of the Wikipedia corpus without MCS.

6.2 Annotating corpora with Babelfy and Nasari

In this section we describe our methodology for disambiguating a multilingual corpus. Our goal is to obtain as many sense annotations as possible, while at the same time retaining high disambiguation accuracy across languages. To this end, we perform a joint disambiguation of both concepts and entities in three successive stages, using BabelNet as reference sense inventory. Our disambiguation strategy is based on three steps: (1) we first construct a multilingual corpus from different resources (Section 6.2.1); (2) we then perform a first high-coverage disambiguation step on this corpus (Section 6.2.2); and, finally, (3) we refine the disambiguation output at the previous step using a procedure based on distributional semantic similarity (Section 6.2.3).

We first apply this method targeting glosses, i.e., textual definitions. Definitions are usually concise and encode "dense", virtually noise-free information that can be best exploited with knowledge acquisition techniques. To date, some of the areas where the use of definitional knowledge has proved to be key in achieving state-of-the-art results are Word Sense Disambiguation [Lesk, 1986, Banerjee and Pedersen, 2002, Navigli and Velardi, 2005, Agirre and Soroa, 2009, Faralli and Navigli, 2012, Fernandez-Ordonez et al., 2012, Chen et al., 2014, Basile et al., 2014, Camacho-Collados et al., 2015b], Taxonomy and Ontology Learning [Velardi et al., 2013, Flati et al., 2016, Espinosa-Anke et al., 2016], Information Extraction [Richardson et al., 1998, Delli Bovi et al., 2015], Plagiarism Detection [Franco-Salvador et al., 2016], and Question Answering [Hill et al., 2015]. The majority of approaches making use of definitions are restricted to corpora where each concept or entity is associated with a single definition; instead, definitions coming from different resources are often complementary and might give different perspectives. Moreover, equivalent definitions of the same concept or entity may vary substantially according to the language, and be more precise or self-explanatory in some languages than others. In fact, the way a certain concept or entity is defined in a given language is sometimes strictly connected to the social, cultural and historical background associated with that language, a phenomenon that also affects the lexical ambiguity of the definition itself. This difference in the degree of ambiguity when moving across languages is especially valuable in the context of disambiguation [Navigli, 2012], as highly ambiguous terms in one language may become less ambiguous (or even unambiguous) in other languages.

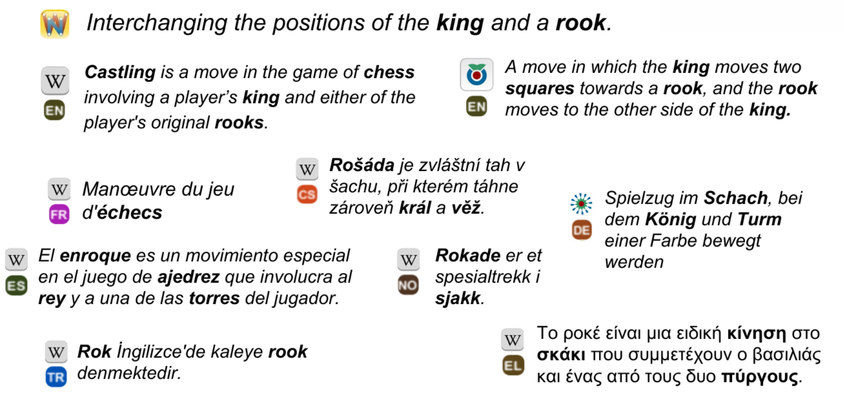
Then, we apply the same method to Europarl [Koehn, 2005][3], one of the most popular multilingual corpora, originally designed to provide aligned parallel text for Machine Translation (MT) systems. Extracted from the proceedings of the European Parliament, the latest release of the Europarl corpus comprises parallel text for 21 European languages, with more than 743 million tokens overall. Apart from its prominent role in MT as a training set, the Europarl corpus has been used for cross-lingual WSD [Lefever and Hoste, 2010, 2013], including, more recently, preposition sense disambiguation [Gonen and Goldberg, 2016], and widely exploited to develop cross-lingual word embeddings [Hermann and Blunsom, 2014, Gouws et al., 2015, Coulmance et al., 2015, Vyas and Carpuat, 2016, Vulić and Korhonen, 2016, Artetxe et al., 2016] as well as multi-sense embeddings [Ettinger et al., 2016, Šuster et al., 2016].

In this section, the key idea is to exploit at best sentences wrote in different languages to provide enriched context for a joint multilingual disambiguation.

6.2.1 Step 1: Harvesting Text in Multiple Languages and Resources

As first step, we need to construct a multilingual corpus. To this end, we first leverage BabelNet, a multilingual lexicalized semantic network obtained from the automatic integration of lexicographic and encyclopedic resources. Thanks to its wide coverage of both lexicographic and encyclopedic terms, BabelNet provides a very large sense inventory for disambiguation, and at the same time a vast and comprehensive target corpus of textual definitions. In fact, as it is a merger of

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



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

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

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

*Manœuvre du jeu d'**échecs***

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

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

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

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

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

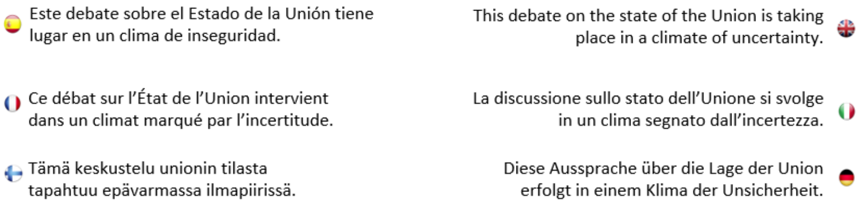
Το ροκέ είναι μια ειδική **κίνηση** στο **σκάκι** που συμμετέχουν ο βασιλιάς και ένας από τους δυο **πύργους**.

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

various different resources, BabelNet provides a heterogeneous set of over 35 million definitions for over 250 languages from WordNet, Wikipedia, Wiktionary, Wikidata and OmegaWiki. To the best of our knowledge, this set constitutes the largest available multilingual corpus of definitional text. Definitional knowledge is not easy to analyze automatically at the sense level. Since many definitions are short and concise, the lack of sufficient and/or meaningful context might negatively affect the performance of an off-the-shelf disambiguation system that works at the sentence level (i.e., targeting individual definitions one by one). In light of this, we leverage the inter-resource and inter-language mappings provided by BabelNet to combine multiple definitions (drawn from different resources and in different languages) of the same concept or entity; in this way, we can associate a much richer context with each target definition, and enable high-quality disambiguation.

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

Then, we moved our focus on translated texts from the web, using Europarl, a corpus of parallel text in 21 languages extracted from the proceedings of the European Parliament. In this case, we identify all available translations of a given sentence and then gather these together into a single multilingual text (see Figure 6.2).



Este debate sobre el Estado de la Unión tiene lugar en un clima de inseguridad.

This debate on the state of the Union is taking place in a climate of uncertainty.

Ce débat sur l'État de l'Union intervient dans un climat marqué par l'incertitude.

La discussione sullo stato dell'Unione si svolge in un clima segnato dall'incertezza.

Tämä keskustelu unionin tilasta tapahtuu epävarmassa ilmapiirissä.

Diese Aussprache über die Lage der Union erfolgt in einem Klima der Unsicherheit.

Figure 6.2. A sentence translated in different languages from Europarl.

6.2.2 Step 2: Context-rich Disambiguation

Once a multilingual text is gathered, an initial preprocessing step is performed. The preprocessing consists of tokenization, part-of-speech (PoS) tagging and lemmatization. We use different preprocessing tools, depending on the language, the polyglot project[4] (a multilingual natural language pipeline), the Stanford CoreNLP pipeline [Manning et al., 2014], the TreeTagger tool [Schmid, 2013] and BABELMORPH[5] (an open-source API based on Wiktionary and designed to retrieve the morphology of content words). Then, we employ Babelfy [Moro et al., 2014b] to disambiguate with high coverage all content words in all the available languages at once. Our methodology is based on the fact that knowledge-based disambiguation systems like Babelfy work better with richer context. In fact, at disambiguation time, Babelfy considers the content words across the target text in order to construct an associated semantic graph, whose richness in terms of nodes and edges strictly depends on the number of content words. As additional text from other resources and languages are included, Babelfy exploits the added context to construct a richer semantic graph. This approach is particularly advantageous for languages with low resources, where standard disambiguation techniques have not yet proven to be effective, due to the lack of sufficient sense-annotated data. As a result of this disambiguation step, we obtain a fully disambiguated corpus, which is later refined by means of distributional semantic similarity. In the following section we explain how this refinement is carried out.

[4] http://polyglot.readthedocs.io/en/latest/index.html

[5] https://github.com/raganato/BabelMorph

6.2.3 Step 3: Disambiguation Refinement based on Distributional Similarity

As output of the previous disambiguation step, we obtained a set D of *disambiguated instances*. These disambiguated instances consist of unambiguous senses from the BabelNet sense inventory, each associated with a confidence score (*Babelfy score* henceforth). However, when the Babelfy score goes below 0.7, a back-off strategy based on the *Most Common Sense* (MCS) is activated by default for that instance. In fact, Babelfy has been shown to be heavily biased towards the MCS (see Chapter 4). At this stage, our task is to reduce this bias by correcting or discarding these low-confidence instances using semantic similarity.

First of all, for each disambiguated instance[6] $d \in D$ we compute a *coherence score* C_d. The coherence score is computed as the number of semantic connections from the BabelNet synset d to any other disambiguated instance in D inside the BabelNet semantic network, divided by the total number of disambiguated instances:

$$C_d = \frac{|\text{Disambiguated instances connected to } d|}{|\text{Disambiguated instances}| - 1} \tag{6.1}$$

We empirically set a coherence score threshold to 0.125 (i.e., one semantic connection out of eight disambiguated instances). Let L be the set of disambiguated instances below both the Babelfy score and the coherence score thresholds (namely the low-confidence annotations). In order to refine the disambiguated instances in L, we use NASARI [Camacho-Collados et al., 2016]. NASARI provides embedded vector representations for over four million BabelNet synsets which were constructed by exploiting the complementary knowledge of Wikipedia, WordNet and text corpora (see Chapter 2). We consider those instances in L for which a NASARI vector can be retrieved (virtually all noun instances), and compute an additional score (*NASARI score*). First, we calculate the centroid μ of all the NASARI vectors for instances in $D \setminus L$. This centroid represents the vector of maximum coherence, as it corresponds to the point in the vector space which is closer to all synsets in D on average. Then, for each disambiguated instance $l \in L$, we retrieve all the candidate senses of its surface form in BabelNet and calculate a NASARI score N_s for each candidate sense. N_s is calculated as the cosine similarity between the

[6] Throughout this step we represent each disambiguated instance as its corresponding synset in BabelNet.

centroid μ and its corresponding NASARI vector $NASARI(s)$:

$$N_s = Sim(\mu, NASARI(s)) \tag{6.2}$$

This score enables us to discard low-confidence disambiguated instances and correct the original disambiguation output from Babelfy in certain cases. Each $l \in L$ is re-tagged with the sense obtaining the highest NASARI score, provided that it exceeds an empirically validated threshold 0.75:

$$\hat{s} = \operatorname*{argmax}_{s \in S_l} N_s \tag{6.3}$$

where S_l is the set containing all the candidate senses for l.

For each corpus we applied this pipeline, we release two versions:

- ***Full.*** This high-coverage version provides sense annotations for all content words as provided by Babelfy after the context-rich disambiguation (see Section 6.2.2), *before* the refinement step.

- ***Refined.*** The refined, high-precision version, instead, *only* includes the most confident sense annotations as computed by the refinement step (see Section 6.2.3).

6.2.4 Building SENSEDEFS

By applying the methodology described on the whole set of textual definitions in BabelNet for all the available languages, we obtain a large multilingual corpus of disambiguated glosses: SENSEDEFS.

Statistics

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

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

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

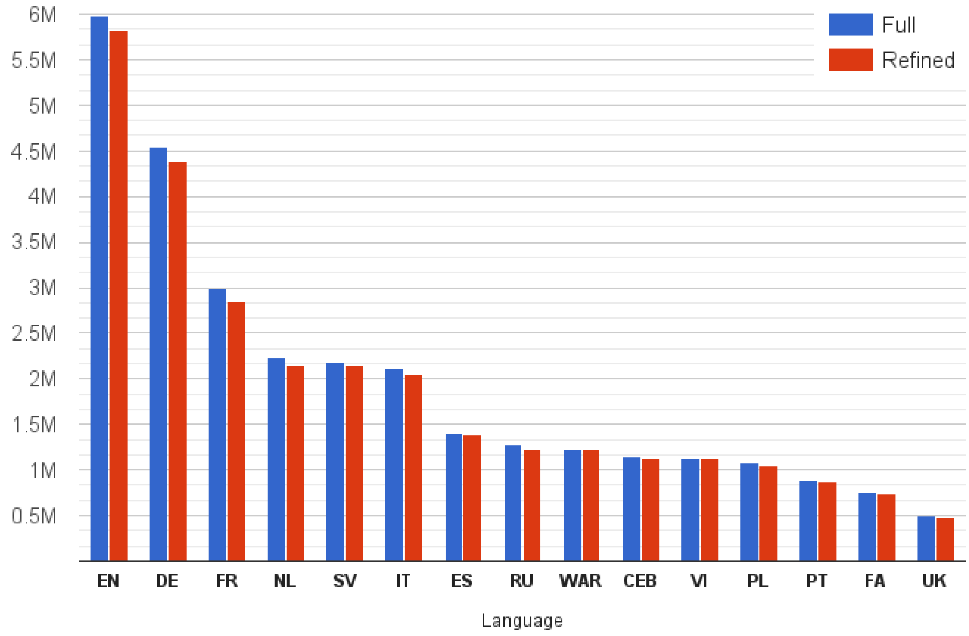


Figure 6.3. Number of definitions by language (top 15 languages).

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

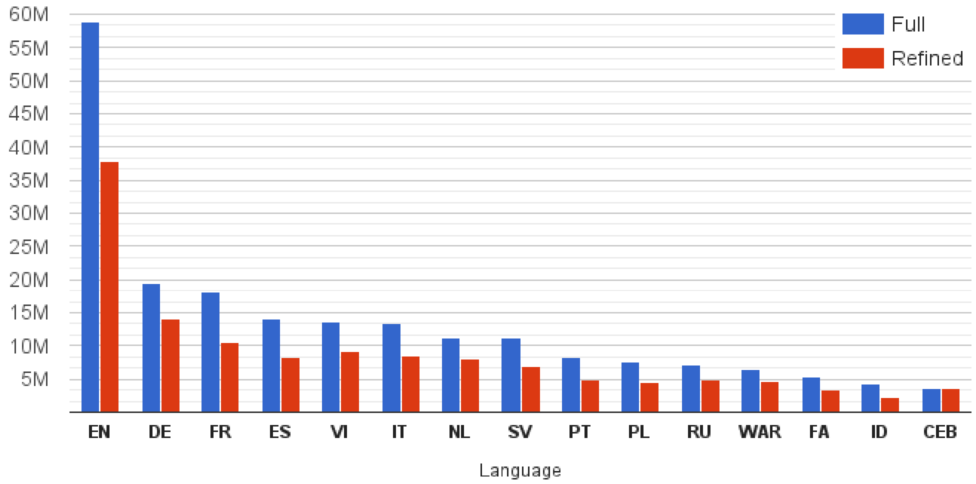


Figure 6.4. Number of annotations by language (top 15 languages).

Statistics by language. Figures 6.3 and 6.4 display the number of definitions and sense annotations, respectively, divided by language[7]. As expected, English provides the largest number of glosses and annotations (5.8M glosses and 37.9M sense annotations in the refined version), followed by German and French. Even though the majority of sense annotations overall concern resource-rich languages (i.e., those featuring the largest amounts of definitional knowledge), the language rankings in Figures 6.3 and 6.4 do not coincide exactly: this suggests, on the one hand, that some languages (such as Vietnamese and Spanish, both with higher positions in Figure 6.4 compared to Figure 6.3) actually benefit from a cross-lingual disambiguation strategy; on the other hand, it also suggests that there is still room for improvement, especially for some other languages (such as Swedish or Russian) where the tendency is reversed and the number of annotations is lower compared to the amount of definitional knowledge available.

Table 6.5 shows the number of annotations divided by part-of-speech tag and disambiguation source. In particular, the full version obtained as output of Step 2 (Section 6.2.2) comprises two disambiguation sources: Babelfy and the MCS back-off (used for low-confidence annotations). The refined version, instead, removes the MCS back-off, either by discarding or correcting the annotation with NASARI (Section 6.2.3). Additionally, 17% of the sense annotations obtained by Babelfy without resorting to the MCS back-off are also corrected or discarded. Assuming

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

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

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

the coverage of the full version to be 100%,[8] the coverage of our system after the refinement step is estimated to be 65.3%. As shown in Table 6.5, discarded annotations mostly consist of verbs, adjectives and adverbs, which are often harder to disambiguate as they are very frequently not directly related to the definitions. In fact, the coverage figure on noun instances is estimated to be 73.9% after refinement.

Evaluation

We evaluated SENSEDEFS both intrinsically and extrinsically on two Natural Language Processing tasks.

Intrinsic Evaluation

As intrinsic evaluation we carried out a thorough manual assessment of sense annotation quality in SENSEDEFS.

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

For each item in the sample, each annotator was shown the textual definition, the BabelNet entry for the definiendum, and every non-MCS sense annotation paired with the corresponding BabelNet entry. The annotator had to decide independently, for each sense annotation, whether it was correct (score of 1), or incorrect (score

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

of 0). The disambiguation source (i.e., whether the annotation came from Babelfy in isolation, context-rich disambiguation or NASARI) was not shown. In some special cases where a certain sense annotation was acceptable but a more suitable synset was available, a score of 0.5 was allowed. One recurrent example of these indecisive annotations occurred on multi-word expressions: being designed as a high-coverage all-word disambiguation strategy, Babelfy can output disambiguation decisions over overlapping mentions when confronted with fragments of text having more than one acceptable disambiguation. For instance, the multi-word expression *"Commission of the European Union"* can be interpreted both as a single mention, referring to the specific BabelNet entity $\texttt{European Commission}_n^1$ (executive body of the European Union), and as two mentions, one (*"Commission"*) referring to the BabelNet entry $\texttt{Parliamentary committee}_n^1$ (a subordinate deliberative assembly), and the other (*"European Union"*) referring to the the BabelNet entry $\texttt{European Union}_n^1$ (the international organization of European countries). In all cases where one part of a certain multi-word expression was tagged with an acceptable meaning, but a more accurate annotation would have been the one associated with the whole multi-word expression, we allowed annotators to assign a score of 0.5 to valid annotations of nested mentions and a score of 1 only to the complete and correct multi-word annotation. Another controversial example of indecision is connected to semantic shifts due to Wikipedia redirections, which cause semantic annotations that are lexically acceptable but wrong from the point of view of semantic roles. For instance, the term *painter* inside Wikipedia redirects to the Wikipedia entry for `Painting` (*Graphic art consisting of an artistic composition made by applying paints to a surface*), while the term *Basketball player* redirects to the Wikipedia entry for `Basketball` (*Sport played by two teams of five players on a rectangular court*). These redirections are also exploited by Babelfy as acceptable disambiguation decisions (a policy that is often used in Entity Linking, especially in Wikipedia-specific settings) and, as such, they are also allowed a score of 0.5.

Once the annotations were completed, we calculated the Inter Annotator Agreement (IAA) between the two annotators of each language by means of Relative Observed Agreement (ROA), calculated as the proportion of equal answers, and Cohen's kappa [Cohen, 1968, κ]. Finally, the two annotators in each language adjudicated the answers which were judged with opposite values. Table 6.6 shows the results of this manual evaluation. In the four languages, our refined version

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

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

of the corpus achieved the best overall results. SenseDefs achieved over 80% precision in three of the four considered languages, both in its full and refined versions. For Italian the precision dropped to 73.9% and 77.5%, respectively, probably due to its lower coverage in BabelNet. Finally, it is worth noting that, for all the examined languages, both the full and refined versions of SenseDefs provided more annotations than using the Babelfy baseline on isolated definitions.

To complement the manual intrinsic evaluation, we performed an additional large-scale automatic evaluation. We compared the WordNet annotations given by SenseDefs [9] with the manually-crafted annotations of the disambiguated glosses from the Princeton Gloss Corpus[10]. Similarly to the previous manual evaluation, we included a baseline based on Babelfy disambiguating the definitions sentence-wise in isolation and using the pre-trained models [11] of the IMS [Zhong and Ng, 2010] supervised disambiguation system. As in our previous experiment, we did not

[9] Our disambiguation pipeline annotates with BabelNet synsets, hence its coverage is larger than only WordNet. This implies that some annotations are not comparable to those inside the WordNet glosses.

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

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

considered the annotations for which the MCS back-off strategy was activated on any of the comparison systems. Finally, as baseline we include the results of WordNet first sense (i.e., MCS) for the annotations disambiguated by each system. The MCS baseline has been shown to be hard to beat, especially for knowledge-based systems (see Chapter 4). However, this baseline, which is computed from a sense-annotated corpus, is only available for the English WordNet. Therefore, it is not possible to use this MCS baseline accurately for languages other than English, and resources other than WordNet for which sense-annotated data is not available or is very scarce.

Table 6.7 shows the accuracy results (computed as the number of annotations corresponding to the manual annotations divided by the total number of overlapping annotations) of SENSEDEFS, Babelfy and IMS on the Princeton Gloss Corpus. SENSEDEFS achieved an accuracy of 76.4%, both in its full and refined versions. Nevertheless, the refined version attained a larger coverage, disambiguating a larger amount of instances. This result is relatively high considering the nature of the corpus, consisting of short and concise definitions for which the context is clearly limited. In fact, even if not directly comparable, the best systems in standard WSD SemEval competitions (where full documents are given as context to disambiguate) tend to obtain considerably less accurate results [Edmonds and Cotton, 2001, Snyder and Palmer, 2004, Pradhan et al., 2007, Navigli et al., 2013, Moro and Navigli, 2015]. In fact, even though results are not directly comparable[12], IMS achieved an accuracy which is considerably lower than our system's performance and also lower compared to its performance on standard benchmarks (see Chapter 4). This result highlights the added difficulty of disambiguating definitions, as they do not provide enough context for an accurate disambiguation in isolation. Only our disambiguation pipeline, which does not make use of any sense-annotated data, proves reliable in this experiment, comfortably outperforming the MCS baseline on the same annotations.

Extrinsic Evaluation

We also evaluated extrinsically the effectiveness of SENSEDEFS (both the *full* and *refined* versions of the resource) by making use of its sense annotations within two Natural Language Processing tasks.

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

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

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

The first experiment evaluated the full version of SENSEDEFS (before refinement) on Open Information Extraction (OIE). The experiment uses DEFIE [Delli Bovi et al., 2015], an OIE system designed to work on textual definitions. In its original implementation DEFIE used Babelfy to disambiguate definitions one-by-one before extracting relation instances. We modified that implementation and used the disambiguated glosses as obtained with our approach as input for the system, and then we compared the extractions with those obtained by the original implementation.

The second experiment, instead, evaluated the refined version of SENSEDEFS on the Sense Clustering task. For this experiment we used the semantic representations of NASARI. In particular, we reconstructed the vectorial representations of NASARI by, 1) enriching the semantic network used in the original implementation with the refined sense annotations of SENSEDEFS, and 2) running again the NASARI pipeline to generate the vectors. We then evaluated these on the Sense Clustering task.

Open Information Extraction. In this experiment we investigated the impact of our disambiguation approach on the definitional corpus used as input for the pipeline of DEFIE. The original OIE pipeline of the system takes as input an unstructured corpus of textual definitions, which are then preprocessed one-by-one to extract syntactic dependencies and disambiguate word senses and entity mentions. After this preprocessing stage, the algorithm constructs a syntactic-semantic graph representation for each definition, from which subject-verb-object triples (relation instances) are eventually extracted. As highlighted in Section 6.2.2, poor context of particularly short definitions may introduce disambiguation errors in the preprocessing stage, which then tend to propagate and reflect on the extraction

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

Table 6.8. Extractions of DEFIE on the evaluation sample.

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

Table 6.9. Precision of DEFIE on the evaluation sample.

of both relations and relation instances. To assess the quality of our disambiguation strategy as compared to the standard approach, we modified the implementation of DEFIE to consider our disambiguated instances instead of executing the original disambiguation step, and then we evaluated the results obtained at the end of the pipeline in terms of quality of relation and relation instances.

Experimental setup. We first selected a random sample of 150 textual definitions from our disambiguated corpus. We generated a baseline for the experiment by discarding all disambiguated instances from the sample, and treating the sample itself as an unstructured text of textual definitions which we used as input for DEFIE, letting the original pipeline of the system carry out the disambiguation step. Then we carried out the same procedure using, instead, the modified implementation for which our disambiguated instances are taken into account. In both cases, we ran the extraction algorithm of DEFIE and evaluated the output in terms of both relations and relation instances. Following Delli Bovi et al. [2015], we employed two human judges and performed the same evaluation procedure described therein over the set of distinct relations extracted from the sample, as well as the set of extracted relation instances.

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

Sense Clustering. This experiment focused on the sense clustering task. Knowledge resources such as Wikipedia or WordNet suffer from the high granularity of their sense inventories. A meaningful cluster of senses within these sense inventories could help boost the performance in different applications [Hovy et al., 2013, Pilehvar et al., 2017]. In the following we explain how to deal with this issue in Wikipedia.

Our method for clustering senses in Wikipedia was based on the semantic representations of NASARI [Camacho-Collados et al., 2016]. We integrated the high-precision version of the network as an enrichment of the BabelNet semantic network, in order to improve the results of the state-of-the-art system based on the NASARI lexical vectors. NASARI uses Wikipedia ingoing links and the BabelNet taxonomy in the process of obtaining contextual information for a given concept. We simply enriched the BabelNet taxonomy with the refined version of the disambiguated glosses of the target language. These disambiguated glosses contain synsets that are highly semantically connected with the definiendum, which makes them particularly suitable for enriching a semantic network. The rest of the pipeline for obtaining lexical semantic representations (i.e., lexical specificity applied to the contextual information) remained unchanged. By integrating the high-precision disambiguated glosses into the NASARI pipeline, we obtained a new set of vector representations for BabelNet synsets, increasing its initial coverage (4.4M synsets covered by the original NASARI, compared to 4.6M synsets covered by NASARI enriched with our disambiguated glosses).

Experimental setup. We used the two sense clustering datasets constructed by Dandala et al. [2013]. In these datasets sense clustering is viewed as a binary classification task. Given a pair of Wikipedia articles, the task consists of deciding whether they should be merged into a single cluster or not. The first dataset (*500-pair* henceforth) contains 500 pairs of Wikipedia articles, while the second dataset (*SemEval*) consists of 925 pairs coming from a set of highly ambiguous words taken from WSD SemEval competitions [Mihalcea, 2007]. We followed the original setting of Camacho-Collados et al. [2016] and clustered a pair of Wikipedia articles only when their similarity, computed by using the square-rooted Weighted Overlap comparison measure [Pilehvar et al., 2013], was above 0.5 (i.e., the middle point in the Weighted Overlap similarity scale).

	500-pair		**SemEval**	
	Acc.	**F1**	**Acc.**	**F1**
NASARI+SenseDefs	**86.0**	**74.8**	**88.1**	**64.7**
NASARI	81.6	65.4	85.7	57.4
SVM-monolingual	77.4	-	83.5	-
SVM-multilingual	84.4	-	85.5	-
Baseline	28.6	44.5	17.5	29.8

Table 6.10. Accuracy (Acc.) and F-Measure (F1) percentages of different systems on the Wikipedia sense clustering datasets.

Results. Table 6.10 shows the accuracy and F1 results in the sense clustering task. As a comparison we included the Support Vector Machine classifier of Dandala et al. [2013], which exploits information from Wikipedia in English (*SVM-monolingual*) and four different languages (*SVM-multilingual*). As a simple baseline we additionally included a system which clusters all pairs. Finally, we report the results of the original NASARI English lexical vectors (*NASARI*[13]) and the NASARI-based vectors obtained from the enriched BabelNet semantic network (*NASARI+SenseDefs*). As shown in Table 6.10, the enrichment produced by our glosses proved to be highly beneficial, significantly improving on the original results obtained by NASARI. Moreover, NASARI+SenseDefs obtained the best performance overall, outperforming the SVM-based systems of Dandala et al. [2013] in terms of accuracy in both datasets.

6.2.5 Building EUROSENSE

Following the the pipeline described in Section 6.2, we augment Europarl with sense-level information for multiple languages: EUROSENSE.

Corpus and Statistics

Table 6.11 reports general statistics on EUROSENSE regarding both its high-coverage (cf. Section 6.2.2) and high-precision (cf. Section 6.2.3) versions. Joint multilingual

[13] Downloaded from `http://lcl.uniroma1.it/nasari/`

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

Table 6.11. General statistics on EuroSense before *(full)* and after refinement *(refined)* for all the 21 languages. Language-specific figures are also reported for the 4 languages of the intrinsic evaluation.

disambiguation with Babelfy generated more than 215M sense annotations of 247k distinct concepts and entities, while similarity-based refinement retained almost 123M high-confidence instances (56.96% of the total), covering almost 156k distinct concepts and entities. 42.40% of these retained annotations were corrected or validated using distributional similarity. As expected, the distribution over parts of speech is skewed towards nominal senses (64.79% before refinement and 81.79% after refinement) followed by verbs (19.26% and 12.22%), adjectives (11.46% and 5.24%) and adverbs (4.48% and 0.73%). We note that the average coherence score increases from 0.19 to 0.29 after refinement, suggesting that distributional similarity tends to favor sense annotations that are also consistent across different languages. Table 6.11 also includes language-specific statistics on the 4 languages of the intrinsic evaluation, where the average lexical ambiguity ranges from 1.12 senses per lemma (German) to 2.26 (English) and, as expected, decreases consistently after refinement.

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

	EN		FR		DE		ES	
	Prec.	**Cov.**	**Prec.**	**Cov.**	**Prec.**	**Cov.**	**Prec.**	**Cov.**
Babelfy	76.1	100	59.1	100	80.4	100	67.5	100
EUROSENSE (full)	80.3	100	67.9	100	84.6	100	76.7	100
EUROSENSE (refined)	**81.5**	75.0	**71.8**	63.5	**89.3**	53.8	**82.5**	62.9

Table 6.12. Precision *(Prec.)* and coverage *(Cov.)* of EUROSENSE, manually evaluated on a random sample in 4 languages. Precision is averaged between the two judges, and coverage is computed assuming each content word in the sense inventory to be a valid disambiguation target.

Experimental Evaluation

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

Intrinsic Evaluation: Annotation Quality

In order to assess annotation quality directly, we carried out a manual evaluation on 4 different languages (English, French, German and Spanish) with 2 human judges per language. We sampled 50 random sentences across the subset of sentences in EUROSENSE featuring a translation in all 4 languages, totaling 200 sentences overall.

For each sentence, we evaluated all sense annotations both before and after the refinement stage, along with the sense annotations obtained by a baseline that disambiguates each sentence in isolation with Babelfy. Overall, we manually verified a total of 5818 sense annotations across the three configurations (1518 in English, 1564 in French, 1093 in German and 1643 in Spanish). In every language the two judges agreed in more than 85% of the cases, with an inter-annotator agreement in terms of Cohen's kappa [Cohen, 1960] above 60% in all evaluations (67.7% on average).

Results, reported in Table 6.12, show that joint multilingual disambiguation improves consistently over the baseline. The similarity-based refinement boosts precision even further, at the expense of a reduced coverage (whereas both Babelfy and the baseline attempt an answer for every disambiguation target). Over the

	SemEval-2013	**SemEval-2015**
$\text{IMS}_{\text{SemCor}}$	65.3	69.3
$\text{IMS}_{\text{OMSTI}}$	65.0	69.1
$\text{IMS}_{\text{EuroSense}}$	**66.4**	**69.5**
UKB	59.0	61.2
UKB_{w2w}	62.9	63.3
MCS	63.0	67.8

Table 6.13. F-Score on all-words WSD.

4 languages, sense annotations appear to be most reliable for German, which is consistent with its lower lexical ambiguity on the corpus.

Extrinsic Evaluation: Word Sense Disambiguation

We additionally carried out an extrinsic evaluation of EUROSENSE by using its refined sense annotations for English as a training set for a supervised all-words WSD system, It Makes Sense [Zhong and Ng, 2010, IMS]. Following Taghipour and Ng [2015a], we started with SemCor [Miller et al., 1993] as initial training dataset, and then performed a subsampling of EUROSENSE up to 500 additional training examples per word sense. We then trained IMS on this augmented training set and tested on the two most recent standard benchmarks for all-words WSD: the SemEval-2013 task 12 [Navigli et al., 2013] and the SemEval-2015 task 13 [Moro and Navigli, 2015] test sets. As baselines we considered IMS trained on SemCor only and OMSTI, the sense-annotated dataset constructed by Taghipour and Ng [2015a] which also includes SemCor. Finally, we report the results of UKB, a knowledge-based system [Agirre et al., 2014].[14] As shown in Table 6.13, IMS trained on our augmented training set consistently outperforms all baseline models, showing the reliability of EUROSENSE as training corpus, even against sense annotations obtained semi-automatically [Taghipour and Ng, 2015a].

6.3 Annotating corpora with hyperlink propagation

In this section we describe our pipeline to augment Wikipedia with as much semantic information as possible, by recovering potentially linkable mentions not covered

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

by original hyperlinks. To achieve this, we rely only on the structure of Wikipedia itself, with no need for recourse to an off-the-shelf disambiguation system. Our approach for building a Semantically Enriched Wikipedia (SEW) takes as input a Wikipedia dump and outputs a sense-annotated corpus, built upon the original Wikipedia text, where mentions are annotated according to the sense inventory of BabelNet [Navigli and Ponzetto, 2012]. Our pipeline applies some standard preprocessing in the first place, including tokenization, part-of-speech tagging and lemmatization. Disambiguation pages, '`List of`' articles and pages of common surnames are discarded, as they typically contain only few lines of meaningful text and introduce noise into the propagation process. After preprocessing, we apply a cascade of *hyperlink propagation heuristics* to the corpus (Section 6.3.1). At each step a different heuristic is applied, enabling our algorithm to identify a list of synsets S^p to be propagated across a given Wikipedia page p; then, for each synset $s \in S^p$, occurrences of any lexicalization of s are detected and added as new annotations for p. All heuristics share a common assumption: given an ambiguous mention within a Wikipedia page, every occurrence of that mention refers to the same sense (*one sense per page*) and hence it is annotated with the same synset. Albeit simple, this assumption is surprisingly accurate[15] and increases coverage substantially.

As we apply a heuristic h to a given Wikipedia page p, we characterize h as being either *intra-page* (when it propagates synsets that occur as mentions within p itself) or *inter-page* (when it exploits the connections of p with other pages or categories). Also, we refer to the *scope* of h as either Wikipedia (when all synsets propagated by h identify a specific Wikipedia page) or BabelNet (when h propagates synsets that may not have an associated Wikipedia page).

After all heuristics have been applied we enforce a conservative policy to remove overlapping mentions and duplicates (i.e., multiple annotations associated with the exact same fragment of text). We deal with overlaps by penalizing inter-page annotations in favor of intra-page ones, and by preferring the longest match in case of overlapping annotations of the same type. Similarly, we deal with duplicates by preferring intra-page annotations over inter-page ones and, if the mention is still ambiguous, we remove *all* its annotations. In other words, we do not attempt to annotate mentions that retain ambiguity even in the context of the same page (and

[15] 98% of the Wikipedia pages support this assumption according to the estimate of Wu and Giles [2015]

	Symbol	Heuristic Type	Scope
Original Hyperlink	HL	-	Wikipedia
Surface Mention Propagation	SP	Intra-page	Wikipedia
Lemmatized Mention Propagation	LP	Intra-page	Wikipedia
Person Mention Propagation	PP	Intra-page	Wikipedia
Wikipedia Inlink Propagation	WIL	Inter-page	Wikipedia
BabelNet Inlink Propagation	BIL	Inter-page	BabelNet
Category Propagation	CP	Inter-page	Wikipedia
Monosemous Content Word	MP	-	BabelNet

Table 6.14. Summary of sense annotation types

connected pages). The set of annotation types is summarized in Table 6.14, while Section 6.3.1 describes each propagation heuristic in detail.

6.3.1 Propagation Heuristics

Intra-page Propagation Heuristics

Intra-page propagation heuristics collect a list of synsets S^p from the original hyperlinks across a Wikipedia page p (including the synset associated with p itself) and then propagate S^p by looking for potential mentions matching any lexicalization of a synset in S^p. Any mention discovered this way is then added to the list of sense annotations for p if part-of-speech tags are consistent. However, as potential mentions may contain punctuation or occur in some inflected form, propagation is performed as a two-pass procedure: a *surface mention propagation* (SP) over the original text of p before preprocessing, and a *lemmatized mention propagation* (LP) over tokenized and lemmatized text. Moreover, as people are not typically referred to by their full name inside the text of an article, we designed a specific heuristic to propagate *person mentions* (PP). If a synset $s \in S^p$ identifies a person according to the BabelNet entity typing, we allow potential mentions to match lexicalizations of s partially (i.e., only first name, or only last name). Each partial mention is then validated by checking surrounding tokens against a precomputed set of first and last names, and added as annotation only if surrounding tokens do not match any person name. This allows us to avoid annotating false positives (e.g., siblings of s).

Inter-page Propagation Heuristics

Inter-page heuristics exploit the connections of p inside Wikipedia and BabelNet. Once synsets to be propagated are collected in S^p, we apply the same propagation procedure. We exploited three inter-page heuristics:

Wikipedia Inlink Propagation (WIL) collects ingoing links to p inside Wikipedia (i.e., other Wikipedia pages where p is mentioned and hyperlinked) and adds the corresponding BabelNet synsets to S^p;

BabelNet Inlink Propagation (BIL), similarly to WIL, leverages ingoing links to the synset s_p that contains p in the BabelNet semantic network. These include, in particular, hyperlinks inside Wikipedias in languages other than English, as well as connections of s_p drawn from other resources integrated in BabelNet;

Category Propagation (CP) propagates hyperlinks across pages that belong to the same Wikipedia categories of p. Intuitively, pages belonging to the same categories tend to mention the same entities. Given a category c, we first harvest all hyperlinks appearing in all Wikipedia pages in c at least twice, and then we rank them by frequency count. In order to filter out categories that are too broad or uninformative (e.g., `Living people`) we associate with each category c a probability distribution over hyperlinks f^c, and compute the entropy $H(c)$ of such distribution as:

$$H(c) = -\sum_{h \in S^c} f^c(h)\, log_2\, f^c(h) \tag{6.4}$$

where h ranges over the set S^c of hyperlinks propagated through category c and $f^c(h)$ is computed as the normalized frequency count of h in S^c. Ranking categories by their entropy values allows us to discriminate between broader categories, where a large number of less related hyperlinks appear with relatively small counts (hence higher H), and more specific categories, where fewer related hyperlinks occur with relatively higher counts (and lower H). Given a Wikipedia page p, we consider each category c_p of p where $H(c_p)$ is below a predefined threshold ρ_H[16], and add to S^p all the synsets that identify hyperlinks in S^{c_p}.

Finally, in order to cover non-nominal content words, we apply a *Monosemous Content Word* (MP) heuristic to propagate verb, adjective and adverb senses that are monosemous according to our sense inventory.

[16] we used $\rho_H = 0.5$ in our experiments (Section 6.3.3)

	# Annotations	# Senses	# Documents	Ann. Type
Wikipedia	71 457 658	2 898 503	4 313 373	Wikipedia
SEW (all)	250 325 257	4 098 049	4 313 373	BabelNet
SEW	206 475 360	4 071 902	4 313 373	BabelNet
SEW-WordNet	116 079 163	67 774	4 313 373	WordNet
SEW-Wikipedia	162 614 753	4 020 979	4 313 373	Wikipedia
Wikilinks	40 323 863	2 933 659	10 893 248	Wikipedia
FACC1	11 240 817 829	5 114 077	1 104 053 884	Freebase
MUN	1 357 922	31 956	62 815	WordNet
MASC	286 416	23 175	392	BabelNet

Table 6.15. Comparison of different sense-annotated corpora. Wikipedia (*first row*) refers to the November 2014 dump.

	Nouns	Verbs	Adjectives	Adverbs
SEW (all)	201 885 731	6 381 452	25 102 343	16 955 731
SEW (conservative)	162 674 740	5 987 696	20 923 743	16 889 181
MUN	687 871	412 482	251 362	6 207
MASC	131 688	82 489	30 015	23 685

Table 6.16. Sense annotations by part of speech

6.3.2 Statistics

We built SEW by applying the approach described in Section 6.3.1 to the English Wikipedia dump of November 2014. We relied on BabelNet [17] as sense inventory, and exploited the Stanford CoreNLP pipeline[18] for preprocessing. Table 6.15 reports some general statistics: the original dump constitutes by itself a corpus of 4,313,373 Wikipedia pages with 71,457,658 sense annotations, covering 2,898,503 distinct synsets. SEW achieves 3.5 times the amount of annotations (58.03 average annotations per page against 16.57 of the original Wikipedia) and adds 1,199,546 new entities not covered by the original hyperlinks. 17.5% ambiguous annotations are removed by our conservative policy, but the overall synset coverage remains almost unchanged. Table 6.15 also includes statistics on SEW with only Wikipedia annotations (fifth row) and only WordNet annotations (fourth row).

The bottom rows of Table 6.15 report comparative statistics on other sense-annotated corpora: Wikilinks [Singh et al., 2012], FACC1 [Gabrilovich et al., 2013], the sense-annotated MultiUN corpus [Taghipour and Ng, 2015a] and the sense-annotated MASC corpus [Moro et al., 2014a]. Compared to Wikilinks, which provides more than 40M annotations from over 10M web pages, the Wikipedia

[17] `http://babelnet.org`

[18] `http://stanfordnlp.github.io/CoreNLP`

	HL	SP	LP	PP	WIL	BIL	CP	MP
SEW (all)	71 457 020	33 780 057	24 510 995	6 735 336	7 237 505	32 713 194	25 650 945	48 240 205
SEW (conservative)	71 457 020	33 589 710	14 936 540	6 411 877	2 174 818	19 850 111	14 271 461	43 783 185

Table 6.17. Sense annotations by annotation type

	SEW (%)	Only HL (%)
Nouns	227 326 282 (38.75%)	116 342 382 (19.83%)
Verbs	8 080 280 (6.71%)	1 799 680 (0.82%)
Adjectives	33 402 556 (27.87%)	9 913 634 (8.27%)
Adverbs	17 163 713 (33.95%)	245 468 (0.49%)
Total	**285 972 831 (29.26%)**	**128 301 164 (13.13%)**

Table 6.18. Coverage of content words by part of speech

portion of SEW adds 122M annotations and 1,087,320 covered senses. FACC1 is considerably larger than any other reported corpus and features 1.12G annotations, which are, however, drawn from 1.1G documents (with an average of 10.18 annotations per document) and restricted to named entities in Freebase. Finally, compared to the sense-annotated MultiUN (MUN) corpus, the WordNet portion of SEW adds over 114M annotations and 35818 covered senses.

Table 6.16 shows sense annotations by part of speech before and after applying the conservative policy. Most annotations are nouns (80.65%), followed by adjectives (10.03%), adverbs (6.77%) and verbs (2.55%). Proportions are somewhat skewed with respect to other corpora, such as MultiUN (50.65% of noun annotations) and the MASC corpus (45.97%), since we include non-noun annotations only when monosemous in our sense inventory.

Table 6.17 shows sense annotations by heuristic type for both intra-page heuristics (above) and inter-page heuristics (below). Each heuristic is identified by the corresponding names in Table 6.14. Apart from original hyperlinks (which provide 28.55% of the annotations) and monosemous mentions (19.27%), the Surface Mention Propagation (SP) and the BabelNet Inlink Propagation (BIL) heuristics provide 13.49% and 13.07% of annotations respectively, followed by the Category Propagation (CP) heuristic with 10.25%. As expected, annotations discarded after applying our conservative policy were mostly derived from inter-page heuristics (WIL, BIL, CP) which open up to a broader context with respect to intra-page ones (and are therefore prone to noisier propagations).

Finally, Table 6.18 reports the coverage at the word level with respect to the original Wikipedia. Out of 977,203,946 content words in total, our approach annotates with senses 38.75% of the nouns, 6.71% of the verbs, 27.87% of the adjectives, and 33.95% of the adverbs. In comparison, original hyperlinks cover 19.83% of the nouns, 8.27% of the adjectives, and less than 1% of verbs and adverbs. Overall, SEW achieves almost 30% coverage on all parts of speech, improving more than 16% with respect to the original Wikipedia (13.3%) and extending coverage to non-nominal content words (verbs, adverbs, adjectives).

6.3.3 Experiments

We evaluated SEW by carrying out both an intrinsic and an extrinsic evaluation. In the former we compared our sense annotations against those discovered by 3W [Noraset et al., 2014], a Wikipedia-specific system designed to add automatically high-precision hyperlinks to Wikipedia pages; in the latter we used SEW as a training set for Entity Linking and we exploited our propagated hyperlinks to develop Wikipedia-based language-independent vector representations for semantic similarity. In both experiments we compared against a baseline given by the original Wikipedia.

Annotation Quality

We assessed the quality of our sense annotations on a hand-labeled evaluation set of 2,000 randomly selected Wikipedia pages, described in Noraset et al. [2014] and used for training, validating and testing 3W. We first ran our annotation pipeline (Sections 6.3.1) on it and then, following Noraset et al. [2014], we checked the 1530 solvable mentions against the gold standard by mapping our sense annotations from BabelNet synsets to Wikipedia pages. Results are reported in Table 6.19 and compared against 3W[19]: while obtaining a substantially higher recall, our approach manages to keep precision above 93% and achieves an F-score of 62.3% against 47.1% of 3W. It is also worth noting that gold standard mentions, being labeled with Wikipedia pages, do not take parts of speech into account and hence include several adjective mentions (e.g., *American*, *German*) labeled as nouns (*United States*, *Germany*), whereas our approach annotates them with the corresponding correct WordNet adjectives ($American^1_a$, $German^1_a$). If we take these cases into

[19] using the recommended setting with threshold at 0.934

	Precision	Recall	F-score
SEW	0.934	**0.468**	**0.623**
SEW w/o SP	0.907	0.409	0.564
SEW w/o LP	0.914	0.456	0.608
SEW w/o PP	0.916	0.457	0.610
SEW w/o WIL	0.917	0.453	0.607
SEW w/o BIL	0.907	0.413	0.567
SEW w/o CP	0.916	0.415	0.571
SEW w/o MP	0.945	0.458	0.617
3W	**0.989**	0.310	0.471

Table 6.19. Results on the hand-labeled gold standard

account, our annotations achieve 96.5% precision and 64.4% F-score, showing that our propagation heuristics reach a precision level comparable to a trained and tuned high-precision linking system, while at the same time granting a much higher coverage, with an average of 31.3 new annotations per page (Section 6.3.2) against an estimate of 7 added by 3W [Noraset et al., 2014].

We used the same gold standard to perform an ablation test on our propagation heuristics: for each heuristic h, we discarded annotations propagated by h and then repeated the experiment. Results (Table 6.19) show that significant contributions in terms of F-score come from both intra-page propagations (SP, +5.89%) and inter-page ones (BIL and CP, +5.2% and +5.3% respectively).

Extrinsic Evaluation: Entity Linking

We evaluated SEW as a training set for EL using IMS [Zhong and Ng, 2010], a state-of-the-art supervised English all-words WSD system based on Support Vector Machines. We then tested IMS on four datasets: the English portion of the **SemEval-2013** task 12 dataset for multilingual WSD [Navigli et al., 2013] and the English named entity portion of the **SemEval-2015** task 13 dataset for multilingual WSD and EL [Moro and Navigli, 2015], both with Wikipedia annotations; the **MSNBC** dataset [Cucerzan, 2007], with 756 mentions extracted from newswire text and linked to Wikipedia, and the test set of **AIDA-CoNLL** [Hoffart et al., 2011]. Results are shown in Table 6.20 for all datasets in terms of F-score: IMS+SEW and IMS+HL represent IMS trained on SEW and IMS trained only on the original Wikipedia hyperlinks (HL), respectively. We include for each dataset a Most Frequent Sense (MFS) baseline provided by BabelNet, as well as results reported by other state-of-the-art EL systems in the literature: Babelfy [Moro et al., 2014b] and the best

	SemEval-2013	SemEval-2015	MSNBC	AIDA-CoNLL
IMS+SEW	**0.810**	**0.882**	**0.789**	**0.726**
IMS+HL	0.775	0.758	0.695	0.712
MFS	0.802	0.857	0.620	0.535
UMCC-DLSI	0.548	-	-	-
Babelfy	**0.874**	-	-	0.821
DFKI	-	**0.889**	-	-
SUDOKU	-	0.870	-	-
Wikifier	-	-	**0.812**	0.724
M&W	-	-	0.685	**0.823**

Table 6.20. Results in terms of F-score on various WSD/EL datasets

performing system reported in Navigli et al. [2013] for SemEval-2013; the two best performing systems reported in Moro and Navigli [2015] for SemEval-2015; finally, Wikifier [Cheng and Roth, 2013] and Wikipedia Miner [Milne and Witten, 2008] (M&W) for MSNBC and AIDA-CoNLL.

In each dataset, IMS trained on SEW consistently outperforms its baseline version trained on the original Wikipedia; this shows that our propagated hyperlinks lead to more accurate supervised models, adding semantic information that enables IMS to generalize better. Furthermore, the IMS model trained on SEW outperforms the best and second-best systems reported in the SemEval 2013 and 2015 tasks, respectively, putting IMS in line with more recent EL approaches, as well as systems specifically designed to exploit Wikipedia information. This suggests that, in general, our sense-annotated corpus has the potential to improve considerably the performance of Wikipedia-based EL systems.

Extrinsic Evaluation: Semantic Similarity

Another interesting test bed for SEW is provided by vector representations for semantic similarity. In fact, several successful approaches to semantic similarity make explicit use of Wikipedia, from ESA [Gabrilovich and Markovitch, 2007] to NASARI [Camacho-Collados et al., 2016]. Others, like SENSEMBED [Iacobacci et al., 2015], report state-of-the-art results when trained on an automatically disambiguated version of Wikipedia. We argue that SEW constitutes a preferable starting point as compared to the original Wikipedia, both in terms of increased hyperlink connections (in the former case) and in terms of increased sense-annotated mentions (in the latter case). To test this experimentally, we designed two sense-based vector representations built upon our corpus:

- A *Wikipage-based representation* (WB-SEW) where we represented each sense s in our sense inventory as a vector v_s where dimensions are Wikipedia pages. We computed, for each page p, the corresponding component of v_s as the frequency of s appearing as annotation in p;

- A *synset-based representation* (SB-SEW) where we represented each Wikipedia page p as a vector v_p where dimensions are BabelNet synsets. We computed, for each synset s, the corresponding component of v_p as the frequency of s appearing as annotation in p.

We estimated frequencies using both raw counts (RC) and lexical specificity (LS), as in Camacho-Collados et al. [2016]. Then we tested our vectors on the two standard benchmarks available for word similarity: the similarity portion of WordSim-353 (**WS-Sim**) and the noun portion of the SimLex-999 dataset (**SimLex-666**). In both cases we relied on *weighted overlap* [Pilehvar et al., 2013] as similarity measure. Following other sense-based approaches [Pilehvar et al., 2013, Camacho-Collados et al., 2016] we adopted a conventional strategy for word similarity that selects, for each word pair, the closest pair of candidate senses.

		WB-SEW		**SB-SEW**		**WB-HL**		**SB-HL**	
		RC	**LS**	**RC**	**LS**	**RC**	**LS**	**RC**	**LS**
WS-Sim	r	**0.65**	0.64	0.50	0.57	0.58	0.58	0.53	0.52
	ρ	0.69	**0.70**	0.56	0.57	0.59	0.61	0.49	0.51
SimLex-666	r	**0.38**	**0.38**	0.26	0.34	0.32	0.32	0.28	0.31
	ρ	0.40	**0.41**	0.33	0.36	0.31	0.32	0.27	0.27

Table 6.21. Results on the word similarity task in terms of Pearson (r) and Spearman (ρ) correlation to human judgement

Table 6.21 reports our performance in comparison with baseline vectors (WB-HL and SB-HL) computed using only the original Wikipedia hyperlinks. Our vector representations improve consistently over the baseline in both datasets. On WS-Sim, in particular, we obtain higher correlation figures than approaches like ADW [Pilehvar et al., 2013] ($r = 0.63$ and $\rho = 0.67$) and ESA ($r = 0.40$ and $\rho = 0.47$), achieving performances in line with the state of the art.

Moreover, since our vector representations are defined with respect to a multilingual sense inventory, we also tested our best performing model (WB-SEW) on a multilingual benchmark given by the **RG-65** dataset and its translations (Table 6.22), consistently beating the baseline and showing a considerable improvement

		WB-SEW		WB-HL		Word2Vec		Polyglot
		RC	LS	RC	LS	original	retro	
EN	r	0.673	**0.674**	0.619	0.614	-	-	0.51
	ρ	0.608	0.620	0.592	0.592	0.73	**0.77**	0.55
FR	r	0.808	**0.811**	0.773	0.778	-	-	0.38
	ρ	0.755	**0.759**	0.693	0.681	0.47	0.61	0.35
DE	r	**0.639**	**0.639**	0.584	0.580	-	-	0.18
	ρ	0.689	**0.695**	0.637	0.615	0.53	0.60	0.15
ES	r	**0.811**	0.804	0.757	0.740	-	-	0.51
	ρ	**0.815**	0.812	0.764	0.759	-	-	0.56

Table 6.22. Pearson (r) and Spearman (ρ) correlation results for multilingual semantic similarity on the RG-65 dataset

on French, German and Spanish over **Word2Vec**, both the original model[20] and the model retrofitted into WordNet [Faruqui et al., 2015] (**retro**), and pre-trained embedding models in the individual languages from the Polyglot project[21] (**Polyglot**).

	WB-SEW		SB-SEW		WB-HL		SB-HL	
	RC	LS	RC	LS	RC	LS	RC	LS
500-pair	0.668	0.668	**0.707**	0.674	0.671	0.654	0.233	0.186
SemEval	0.630	0.642	0.630	**0.645**	0.562	0.558	0.294	0.239

Table 6.23. F-score results on Wikipedia sense clustering

Finally, we tested our vector representations on the Wikipedia sense clustering task described in [Dandala et al., 2013], evaluating on both benchmark datasets (**500-pair** and **SemEval**). For each sense pair we thus computed similarity as in the previous experiment, and then checked it against empirically validated clustering thresholds of $t = 0.1$ (WB-SEW) and $t = 0.5$ (SB-SEW). Results reported in Table 6.23 are consistent with the experiment on word similarity (Table 6.21) and show that our vector representations improve consistently over their baseline counterparts, with F-scores close to (or slightly above) the state of the art reported by NASARI (72% on 500-pair and 64.2% on SemEval).

[20] we report results of pre-trained vectors over the Google News corpus (EN) and 1 billion tokens from Wikipedia (DE and FR)

[21] https://sites.google.com/site/rmyeid/projects/polyglot

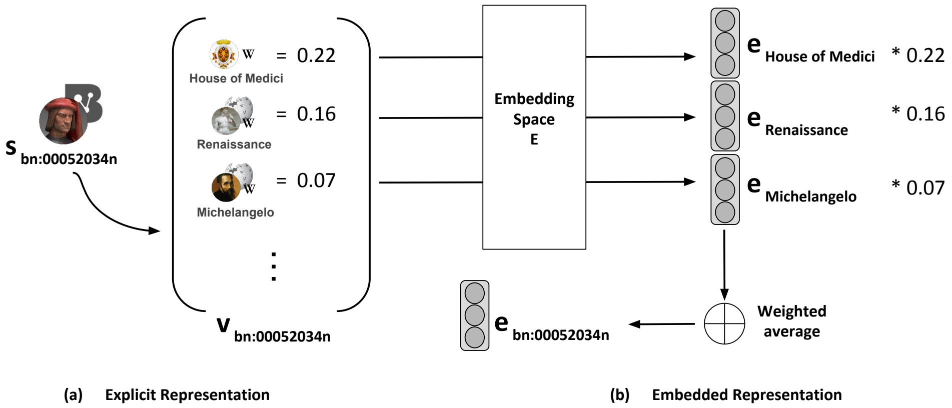


Figure 6.5. Illustrative example of SEW-EMBED's embedded representation (*b*) for the BabelNet entity *Lorenzo de Medici* (`bn:00052034n`) obtained from the corresponding explicit representation (*a*).

6.3.4 Building Vectors from Sense Annotations

In this section we provide the details of SEW-EMBED. The workflow of our procedure is depicted in Figure 6.5 with an illustrative example.

Embedded Representation

In order to compute the embedded augmentation of an explicit vector v_s, obtained as in Section 6.3.3 for a given concept or entity s, we follow Camacho-Collados et al. [2016] and exploit the compositionality of word embeddings [Mikolov et al., 2013b]. According to this property, the representation of an arbitrary compositional phrase can be expressed as the combination (typically the average) of its constituents' representations. We build on this property and plug a pre-trained embedding representation into the explicit representation. In particular, we consider each dimension p (i.e., Wikipedia page) of v_s and map it to the embedding space E provided by the pre-trained representation to obtain an embedded vector e_p. Such mapping depends on the specific embedding representation:

- In case of a *word* embedding representation we consider the Wikipedia page title as lexicalization of p and then retrieve the associated pre-trained embedding. If the title is a multi-word expression and no embedding is available for the whole expression, we exploit compositionality again and average the embedding vectors of its individual tokens;

- In case of a *sense* or *concept* embedding representation we instead exploit

BabelNet's inter-resource links, and map p to the target sense inventory for which the corresponding embedding vector can be retrieved.

The embedded representation e_s of s (Figure 6.5b) is then computed as the weighted average over all the embedded vectors e_p associated with the dimensions of v_s:

$$e_s = \frac{\sum_{p \in v_s} \omega_p \, e_p}{\sum_{p \in v_s} \omega_p} \tag{6.5}$$

where ω_p is the lexical specificity weight of dimension p. In contrast to a simple average, here we exploit the ranking of each dimension p (represented by ω_p) and hence give more importance to the higher weighted dimensions of v_s.

Word Similarity

In order to calculate similarity at the word level, we follow other sense-based approaches [Pilehvar et al., 2013, Camacho-Collados et al., 2016] and adopt a strategy that selects, for a given word pair w_1 and w_2, the *closest* pair of candidate senses:

$$Sim(w_1, w_2) = \max_{s_1 \in S_{w_1},\, s_2 \in S_{w_2}} \sigma(\vec{s_1}, \vec{s_2}) \tag{6.6}$$

where S_w is the set of candidate senses of w in the BabelNet sense inventory, and $\vec{s}$ is the vector representation associated with $s \in S_w$. As similarity measure σ we use standard *cosine similarity* for SEW-EMBED, and *weighted overlap* [Pilehvar et al., 2013] for the explicit representations based on SEW.

Finally, we rely on a back-off strategy that set $Sim(w_1, w_2) = 0.5$ (i.e., the middle point in our similarity scale) when no candidate sense is found for either w_1 or w_2.

Experiments

In this section we report and discuss the performance of SEW-EMBED on the monolingual and cross-lingual benchmark of the SemEval 2017 Task 2 [Camacho Collados et al., 2017]. For completeness we also include the best system of the task, marked with $*$.[22] We consider two versions of SEW-EMBED: one based on the pre-

[22] For an extensive comparison including all participating systems in the task, the reader is referred to the task description paper.

trained word embeddings of Word2Vec [Mikolov et al., 2013a, **SEW-EMBED**$_{w2v}$][23], and another one based on the embedded concept vectors of NASARI [Camacho-Collados et al., 2016, **SEW-EMBED**$_{Nasari}$]. In all test sets, the figures of **SEW-EMBED**$_{w2v}$ correspond to the results of SEW-EMBED reported in the task description paper [Camacho Collados et al., 2017]. We additionally include the results obtained by the original explicit representations based on SEW and by the NASARI baseline, and use them as comparison systems.

	EN			FA			DE			IT			ES		
	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean
SEW-EMBED$_{w2v}$	0.56	0.58	0.57	0.38	0.40	0.39	0.45	0.45	0.45	0.57	0.57	0.57	0.61	0.62	0.62
SEW-EMBED$_{Nasari}$	0.57	0.61	0.59	0.30	0.40	0.34	0.38	0.45	0.42	0.56	0.62	0.59	0.59	0.64	0.62
SEW	0.61	0.67	0.64	**0.51**	**0.56**	**0.53**	**0.51**	**0.53**	**0.52**	**0.63**	**0.70**	**0.66**	**0.60**	**0.66**	**0.63**
NASARI	**0.68**	**0.68**	**0.68**	0.41	0.40	0.41	**0.51**	0.51	0.51	0.60	0.59	0.60	**0.60**	0.60	0.60
Luminoso_run2*	0.78	0.80	0.79	0.51	0.50	0.50	0.70	0.70	0.70	0.73	0.75	0.74	0.73	0.75	0.74

Table 6.24. Results on the multilingual word similarity benchmarks (subtask 1) of Semeval 2017 task 2, in terms of Pearson correlation (r), Spearman correlation (ρ), and the harmonic mean of r and ρ.

Subtask 1: Multilingual Word Similarity

Table 6.24 shows the overall performance on multilingual word similarity for each monolingual dataset. Both **SEW-EMBED**$_{w2v}$ and **SEW-EMBED**$_{Nasari}$ achieve comparable results: their correlation figures are in the same ballpark as the NASARI baseline for Italian, Farsi, and Spanish; instead, they lag behind in English and German. Most surprisingly, however, the explicit representations based on SEW show an impressive performance, and reach the best result overall in 4 out of 5 benchmarks: this might suggest that many word pairs across the test sets are actually being associated with concepts or entities that are well connected in the semantically enriched Wikipedia, and hence the corresponding sparse vectors are representative enough to provide meaningful comparisons. In general, the performance decrease on German and Farsi for all comparison systems is connected to the lack of coverage: both SEW and SEW-EMBED use the back-off strategy 70 times for Farsi (14%) and 54 times (10.8%) for German.

[23] We utilized the pre-trained models available at `https://code.google.com/archive/p/word2vec`. These models were trained on a Google News corpus of about 100 billion words.

	DE-ES			DE-FA			DE-IT			EN-DE			EN-ES		
	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean
SEW-EMBED$_{w2v}$	0.52	0.54	0.53	0.42	0.44	0.43	0.52	0.52	0.52	0.50	0.53	0.51	0.59	0.60	0.59
SEW-EMBED$_{Nasari}$	0.47	0.55	0.51	0.35	0.45	0.39	0.47	0.55	0.51	0.46	0.55	0.50	0.59	0.63	0.61
SEW	**0.57**	**0.61**	**0.59**	**0.53**	**0.58**	**0.56**	**0.59**	**0.64**	**0.61**	0.58	**0.62**	**0.60**	0.61	**0.63**	0.61
NASARI	0.55	0.55	0.55	0.46	0.45	0.46	0.56	0.56	0.56	**0.60**	0.59	**0.60**	**0.64**	**0.63**	**0.63**
Luminoso_run2*	0.72	0.74	0.73	0.59	0.59	0.59	0.74	0.75	0.74	0.76	0.77	0.76	0.75	0.77	0.76

	EN-FA			EN-IT			ES-FA			ES-IT			IT-FA		
	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean	r	ρ	Mean
SEW-EMBED$_{w2v}$	0.46	0.49	0.48	0.58	0.60	0.59	0.50	0.53	0.52	0.59	0.60	0.60	0.48	0.50	0.49
SEW-EMBED$_{Nasari}$	0.41	0.52	0.46	0.59	0.65	0.62	0.44	0.54	0.48	0.58	0.64	0.61	0.42	0.52	0.47
SEW	**0.58**	**0.63**	**0.61**	0.64	**0.71**	**0.68**	**0.59**	**0.65**	**0.62**	**0.63**	**0.70**	**0.66**	**0.59**	**0.65**	**0.62**
NASARI	0.52	0.49	0.51	**0.65**	0.65	0.65	0.49	0.47	0.48	0.60	0.59	0.60	0.50	0.48	0.49
Luminoso_run2*	0.60	0.59	0.60	0.77	0.79	0.78	0.62	0.63	0.63	0.74	0.77	0.75	0.60	0.61	0.60

Table 6.25. Results on the cross-lingual word similarity benchmarks (subtask 2) of SemEval 2017 task 2, in terms of Pearson correlation (r), Spearman correlation (ρ), and the harmonic mean of r and ρ.

Subtask 2: Cross-lingual Word Similarity

Table 6.25 reports the overall performance on cross-lingual word similarity for each language pair. Consistently with the multilingual evaluation, both **SEW-EMBED**$_{w2v}$ and **SEW-EMBED**$_{Nasari}$ achieve comparable results in the majority of benchmarks. All approaches based on SEW seem to perform globally better in a cross-lingual setting: on average, the harmonic mean of r and ρ is 2.2 points below the NASARI baseline. This suggests the potential of Wikipedia as a bridge to multilinguality: in fact, even though SEW was constructed automatically on the English Wikipedia, knowledge transfers rather well via inter-language links and has a considerable impact on the cross-lingual performance.

Again, the best figures are consistently achieved by the explicit representations based on SEW: the improvement in terms of harmonic mean of r and ρ is especially notable in benchmarks that include a less-resourced language such as Farsi (+11.75% on average compared to the NASARI baseline). This improvement does not occur with SEW-EMBED, since in that case sparse vectors are eventually mapped to an embedding space trained specifically on an English corpus.

General Discussion

Overall, SEW-EMBED reached the 4th and 3rd positions in the global rankings of subtask 1 and 2 respectively (with scores 0.552 and 0.558, not including the NASARI baseline). Thus, perhaps surprisingly, the embedded augmentation yielded a considerable decrease in terms of global performance in both subtasks, where the original explicit representations of SEW achieved a global score of 0.615 in subtask 1, and a global score of 0.63 in subtask 2. [24]

Intuitively, multiple factors might have influenced this negative result:

- **Dimensionality Reduction.** Converting an explicit vector (with around 4 million dimensions) into a latent vector of a few hundred dimensions leads inevitably to losing some valuable information, and hence to a decrease in the representational power of the model. Such a phenomenon was also shown by Camacho-Collados et al. [2016], where the lexical and unified representations of NASARI tend to outperform the embedded representation on several word similarity and sense clustering benchmarks;

- **Lexical Ambiguity.** While the original concept vectors of SEW are defined in the unambiguous semantic space of Wikipedia pages, we constructed their embedded counterparts via the word-level representations of their lexicalized dimensions; hence, when moving to the word level, we ended up conflating the different meanings of an ambiguous word or expression;[25]

- **Non-Compositionality.** The compositional properties of word embeddings that we assumed falls short in many cases, such as idiomatic expressions or named entity mentions (e.g., *Wall Street*, or *New York*). The explicit vectors of SEW, instead, do not require the compositional assumption and always consider a multi-word expression as a whole.

Even though the embedded representations of SEW do not match up to the accuracy of explicit ones on experimental benchmarks, they are on the other hand more convenient in terms of compactness and flexibility (due to the reduced dimensionality), and also in terms of comparability, as they are defined in the same vector space of Word2Vec-based representations such as the embedded vectors of NASARI [Camacho-Collados et al., 2016] or DECONF [Pilehvar and Collier, 2016].

[24] The global score is computed as the average harmonic mean of Pearson and Spearman correlation on the best four (subtask 1) and six (subtask 2) individual benchmarks [Camacho Collados et al., 2017].

[25] E.g., in SEW-EMBED$_{w2v}$, the distinct explicit dimensions represented in SEW by the Wikipedia pages BANK and BANK (GEOGRAPHY) were both mapped to the Word2Vec embedding of *bank*.

6.4 Conclusion

In this chapter, we presented different techniques to automatically generate sense annotated text. Starting from disambiguating Wikipedia with a state-of-the-art multilingual knowledge-based disambiguation system, i.e.e Babelfy, obtaining a large corpus sense annotated, we moved towards multilingual text corpora, by leveraging the structure of a wide-coverage semantic network and sense inventory like BabelNet, obtaining a corpus of textual definitions coming from multiple sources and multiple languages, and by using parallel corpora. We developed a pipeline to get higher quality annotations. Our pipeline carries out disambiguation in two subsequent stages. In the first stage, we leverage Babelfy [Moro et al., 2014b], which is designed to exploit at best a multiple-language setting. Using Babelfy, we obtain an initial set of sense annotations for all the available languages of the target corpus. These initial sense annotations are then refined in the second stage, by integrating a module based on NASARI [Camacho-Collados et al., 2016] and distributional similarity targeted to identify a subset of sense annotations disambiguated with high-confidence.

Thanks to our pipeline, we build SENSEDEFS, a corpus of textual definitions coming from multiple sources and multiple languages, and EUROSENSE, a large multilingual sense-annotated corpus based on Europarl. For both corpora, we released a full version comprising all the sense annotations obtained with Babelfy in the first stage, and a refined version including only the high-confidence annotations identified through distributional similarity. Both versions additionally include a set of confidence scores which can be taken into account by users for tuning them to their needs. We evaluated both versions extensively, with both intrinsic and extrinsic experiments, showing the reliability of our system in comparison to previous approaches, leading to performance improvement across different Natural Language Processing tasks.

Moreover, we have presented the automatic construction and evaluation of SEW, a Semantically Enriched Wikipedia, where the overall number of linked mentions has been more than tripled by exploiting at best the hyperlink structure of Wikipedia and the wide-coverage sense inventory of BabelNet. Our approach is surprisingly simple, fully automatic and self contained, with no training, validation or tuning. The extensive evaluation proved the quality of our annotations and that SEW is a flexible resource, suitable for different tasks where our simple benchmark systems are able to set important performance baselines, suggesting its potential for multilingual and cross-lingual applications. To the best of our knowledge, SEW is the largest available resource that comprises word senses and named entity mentions together, annotated using the same sense inventory.

All the built sense annotated corpora are publicly available, and we hope this could pave the way for the designing of more robust multilingual neural models for WSD applied in downstream application [Flekova and Gurevych, 2016, Pilehvar et al., 2017].

Chapter 7

Conclusions and Future Work

In this thesis we addressed the historical task aiming at assigning meanings to word occurrences within text, i.e., Word Sense Disambiguation. Looking at the state of the WSD field, we encountered different problems. We saw that a major issue was the lack of a well-formed framework to perform experiments and analysis. Despite the organizing of the Senseval/SemEval series, providing testing data to the community, the various competitions have few things in common, ranging from the format of the file to the utilized sense inventory. This hampered the development of the WSD field which is currently suffering from lack of real improvements, making hard to draw conclusions on the actual factors which impact the performance of a system.

As primary efforts in this direction, we described the entire workflow of the construction of a unified evaluation framework for WSD (see Chapter 4). Starting from collecting all the datasets from the international competitions Senseval/SemEval, we converted them all to a unified XML format. Then, we semi-automatically mapped the sense inventory of each dataset to WordNet 3.0, adding preprocessing information (e.g., PoS tag and lemma) to each token. The constructed dataset is used to perform an empirical comparison among the major WSD systems, testing both knowledge-based and supervised approaches. Thanks to this framework we are able to make quantitative and qualitative confrontations in a fair setting, on more than 7K test instances. Our experimental analysis shows supervised systems consistently outperform their knowledge-based counterpart. Moreover, enriching the training data with datasets automatically annotated generally helps to boost the performance of supervised systems. The knowledge-based approaches manage to reach good performance for nouns, but they lose ground in the other parts of speech, specially for verbs. One straightforward way to address this issue would be by enriching the semantic network with more cross PoS relations, given that the majority of relations connect words from the same part of speech. We also noticed that each system presents a strong bias towards MFS. Naturally, supervised systems are affected by the bias of their underlying training corpus (even though neural model seems to be less bias), while the semantic network exploited in the knowledge-based systems presents more connections for the MFS candidates [Calvo and Gelbukh, 2015].

As result of our framework and the analysis provided, we moved our focus to the more promising supervised systems, studying the role of different neural sequence learning models for WSD (see Chapter 5). Taking inspiration from previous approaches [Vinyals et al., 2015] we exploited the flexibility of neural models, from Long Short-Term Memory to Encoder-Decoder, showing that we can overcome the word-expert assumption, disambiguating jointly all words in a sentence, retaining state-of-the-art accuracy. Furthermore, we show that this flexibility is such that, for the first time in WSD, a model trained on a given language is able to seamlessly handle a different language at testing time. Our extensive evaluations provide a first solid step to develop more sophisticated neural networks.

Being aware that supervised models tend to perform better, at the expense of requiring huge amount of annotated data, and annotating data is quite an expensive process, we also investigated several ways to get automatically high quality sense annotated data for multiple languages (see Chapter 6). We presented different approaches, from using only a multilingual off-the-shelf knowledge-based system to combining it with semantic similarity to improve the quality of the data. Furthermore, we also developed a method to enrich Wikipedia with as many annotations as possible, without relying on any off-the-shelf system. Our intrinsic and extrinsic evaluation in several tasks proved the high quality of our annotations. Ending up with almost 250 million sense annotations of over 35 million definitions for 256 languages gathered from the wide sense inventory of BabelNet, almost 123 million sense annotations for over 155 thousand distinct concepts and entities in 21 languages from Europarl and more than 200 million annotations of over 4 million different concepts and named entities from the English Wikipedia, we presented the largest available collection of sense annotated corpora.

Finally, we conclude the thesis by mentioning future directions based on this work:

- A direction left to future work is certainly the extension of our unified framework to languages other than English, including SemEval multilingual WSD datasets [Navigli et al., 2013, Moro and Navigli, 2015], as well as to other sense inventories such as BabelNet and Wikipedia, which are available in different languages. The use of other sense inventories of different granularity might be very useful in order to understand how much the ambiguity level impact on a system performance, and how much is the gap between knowledge-based and supervised systems in a coarse-grained sense inventory such as Wikipedia.

- Another research direction would be to explore different approaches to WSD, from multi-modal setting, using images to a graph-augmentation training of neural networks. Images can bring useful complementary information as already proved for other tasks [Calixto et al., 2017, Calixto and Liu, 2017]. Furthermore, neural graph networks [Bui et al., 2017, Chen et al., 2017b] seem very appealing models to study for WSD. These models could be beneficial for new research addressing domain bias, and cross-linguality integrating prior knowledge into a neural network.

- One of the major criticism on WSD is the lack of integration into downstream applications, despite the potential benefits. Few attempts have been made by the community, with varying degrees of success, replacing word embeddings with sense embeddings in multiple tasks such as topic categorization [Li and Jurafsky, 2015, Flekova and Gurevych, 2016, Pilehvar et al., 2017]. Neural machine translation could be another potential downstream application to benefit from WSD. Despite the huge success of neural models, seemingly powerful enough to disambiguate the words in context without relying on a disambiguation pipeline, more ambiguous is the sentence, more likely the system fails [Liu et al., 2017]. Thus, future work could also investigate how to integrate a WSD module into a machine translation system.

- Another direction left to future work would be to explore different approaches to learn multilingual word and sense embeddings in the same space from all our annotated data [Iacobacci et al., 2015, Mancini et al., 2017]. A promising direction regards cross-lingual experiments, we are the first, to the best of our knowledge, to start investigate a multilingual setting for supervised WSD. Being a new area for WSD, it is a research that needs more detailed analysis on each language and further exploration.

- Finally, as concerns the automatically construction of sense annotated corpora, another direction would be to further refine the quality of sense annotations, developing systems able to be applied to multiple languages, covering not only nouns but also verbs, adjectives and adverbs. In particular, future perspectives include the extension of SEW to Wikipedias in other languages, moving towards the construction of a larger, multilingual sense-annotated corpus.

Appendix

SupWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation

Beside the automatic harvesting of sense-annotated data for different languages, a variety of multilingual preprocessing pipelines has also been developed across the years [Padró and Stanilovsky, 2012, Agerri et al., 2014, Manning et al., 2014, Straka and Straková, 2017]. To date, however, very few attempts have been made to integrate these data and tools with a supervised WSD framework; as a result, multilingual WSD has been almost exclusively tackled with knowledge-based systems, despite the fact that supervised models have been proved to consistently outperform knowledge-based ones in all standard benchmarks (see Chapter 4). As regards supervised WSD, It Makes Sense [Zhong and Ng, 2010, IMS] is indeed the de-facto state-of-the-art system used for comparison in WSD, but it is available only for English, with the last major update dating back to 2010.

The publicly available implementation of IMS suffers from different crucial drawbacks: (i) the design of the software makes the current code difficult to extend (e.g., with classes taking as input more than 15 parameters); (ii) the implementation is not optimized for larger datasets, being rather time- and resource-consuming. These difficulties hamper the work of contributors willing to update it, as well as the effort of researchers that would like to use it with languages other than English. For example, in the DKPro WSD framework [Miller et al., 2013], the IMS system is imported as it is, dragging with it the aforementioned drawbacks of the original system. Instead, here, the main purpose is to rebuild entirely the implementation of IMS from scratch.

In this Appendix we present SUPWSD, whose objective is to facilitate the use of a supervised WSD software for both end users and researchers. SUPWSD is designed to be modular and highly flexible, enabling contributors to extend it with ease. Its usage is simple and immediate: it is based on a jar file with only 2 commands and 3 parameters, along with an XML configuration file for specifying customized settings. SupWSD supports the most widely used multilingual preprocessing tools in the research community: Stanford coreNLP [Manning et al., 2014], openNLP[1], TreeTagger [Schmid, 2013] and UDPipe [Straka and Straková, 2017]; as such, SupWSD can directly handle all the languages supported by these tools. Finally, its architecture design relies on commonly used design patterns in Java (such as Factory and Observer among others), which make it flexible for a programmatic use and easily expandable.

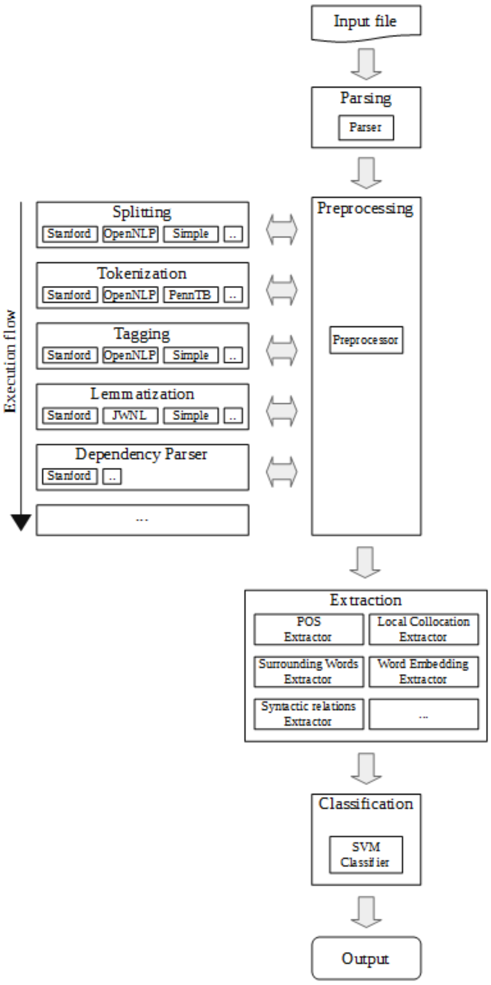
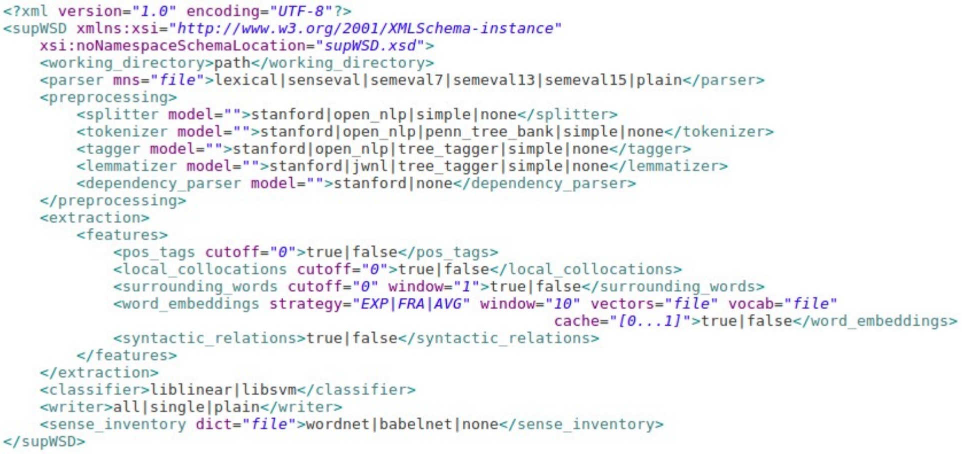


Figure 7.1. Architecture design of SupWSD.

[1] opennlp.apache.org/



```
<?xml version="1.0" encoding="UTF-8"?>
<supWSD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="supWSD.xsd">
    <working_directory>path</working_directory>
    <parser mns="file">lexical|senseval|semeval7|semeval13|semeval15|plain</parser>
    <preprocessing>
        <splitter model="">stanford|open_nlp|simple|none</splitter>
        <tokenizer model="">stanford|open_nlp|penn_tree_bank|simple|none</tokenizer>
        <tagger model="">stanford|open_nlp|tree_tagger|simple|none</tagger>
        <lemmatizer model="">stanford|jwnl|tree_tagger|simple|none</lemmatizer>
        <dependency_parser model="">stanford|none</dependency_parser>
    </preprocessing>
    <extraction>
        <features>
            <pos_tags cutoff="0">true|false</pos_tags>
            <local_collocations cutoff="0">true|false</local_collocations>
            <surrounding_words cutoff="0" window="1">true|false</surrounding_words>
            <word_embeddings strategy="EXP|FRA|AVG" window="10" vectors="file" vocab="file"
                                                            cache="[0...1]">true|false</word_embeddings>
            <syntactic_relations>true|false</syntactic_relations>
        </features>
    </extraction>
    <classifier>liblinear|libsvm</classifier>
    <writer>all|single|plain</writer>
    <sense_inventory dict="file">wordnet|babelnet|none</sense_inventory>
</supWSD>
```

Figure 7.2. The XML configuration file used by SupWSD.

SupWSD: Architecture

In this section we describe the workflow of SupWSD. Figure 7.1 shows the architecture design of our framework: it is composed of four main modules, common for both the training and testing phase: (i) input parsing, (ii) text preprocessing, (iii) features extraction and (iv) classification.

Input parsing. Given either a plain text or an XML file as input, SupWSD first parses the file and extracts groups of sentences to provide them as input for the subsequent text preprocessing module. Sentence grouping is used to parallelize the preprocessing module's execution and to make it less memory-intensive. Input files are loaded in memory using a lazy procedure (i.e., the parser does not load the file entirely at once, but processes it according to the segments of interest) which enables a smoother handling of large datasets. The parser specification depends on the format of the input file via a Factory patterns, in such a way that new additional parsers can easily be implemented and seamlessly integrated in the workflow SupWSD currently features 6 different parsers, targeted to the various formats of the Senseval/SemeEval WSD competition (both all-words and lexical sample), along with a parser for plain text.

Text preprocessing. The text preprocessing module runs the pre-specified preprocessing pipeline on the input text, all the way from sentence splitting to dependency parsing, and retrieves the data used by the feature extraction module to construct the features. This module consists of a five-step pipeline: sentence splitting, tokenization, part-of-speech tagging, lemmatization and dependency parsing. SUPWSD currently supports three preprocessing options: Stanford, UDPipe and Hybrid. They can be switched on and off using the configuration file. The fist two provide a wrapper for the Stanford NLP and UDPipe pipeline respectively, and select the default model for each component. The latter, instead, enables the user to customize their model choice for each and every preprocessing step. For instance, one possible customization is to use the openNLP models for tokenization and sentence splitting, and the Stanford models for part-of-speech tagging and lemmatization. In addition, the framework enables the user to provide an input text where preprocessing information is already included.

The communication between the input parsing and the text preprocessing modules (Figure 7.1) is handled by the `Analyzer`, a component that handles a fixed thread pool and outputs the feature information collected from the input text.

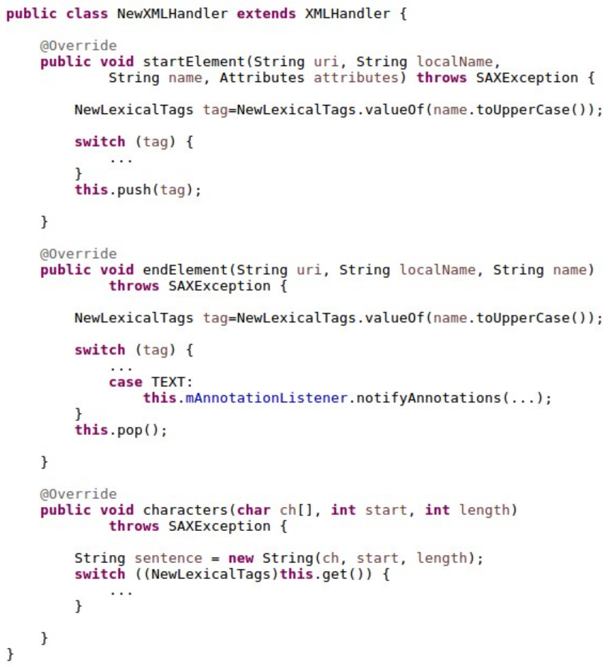
Features extraction. The feature extraction module takes as input the data extracted at preprocessing time, and constructs a set of features that will be used in the subsequent stage to train the actual SUPWSD model. As in the previous stage, the user can rely on the configuration file (Figure 7.2) to select which features to enable or disable. SUPWSD currently supports five standard features: (i) *part-of-speech tag* of the target word and part-of-speech tags surrounding the target word (with a left and a right window of length 3); (ii) *surrounding words*, i.e., the set of word tokens (excluding stopwords from a pre-specified list) appearing in the context of the target word; (iii) *local collocations*, i.e., ordered sequences of tokens around the target word; (iv) pre-trained *word embedding*, integrated according to three different strategies, as in Iacobacci et al. [2016];[2] (v) *syntactic relations*, i.e., a set of features based on the dependency tree of the sentence, as in Lee and Ng [2002]. SUPWSD allows the user to select appropriate cutoff parameters for features (i) to (iii), in order to filter them out according to a minimum frequency threshold.

Classification. The classification module constitutes the last stage of the SUPWSD pipeline. On the basis of the feature set constructed in the previous stage, this module leverages an off-the-shelf machine learning library to run a classification algorithm and generate a model for each sense-annotated word type in the input

[2] We implemented a cache mechanism in order to deal efficiently with large word embedding files.

text. The current version of SUPWSD relies on two widely used machine learning frameworks: LIBLINEAR[3] and LIBSVM[4]. The classification module of SUPWSD operates on top of these two libraries.

Using the configuration file (Figure 7.2) the user can select which library to use and, at the same time, choose the underlying sense inventory. The current version of SUPWSD supports two sense inventories: WordNet [Miller, 1995][5] and BabelNet [Navigli and Ponzetto, 2012][6]. Specifying a sense inventory enables SUPWSD to exploit the Most Frequent Sense (MFS) back-off strategy at test time for those target words for which no training data are available.[7] If no sense inventory is specified, the model will not provide an answer for those target words.



```
public class NewXMLHandler extends XMLHandler {

    @Override
    public void startElement(String uri, String localName,
            String name, Attributes attributes) throws SAXException {

        NewLexicalTags tag=NewLexicalTags.valueOf(name.toUpperCase());

        switch (tag) {
            ...
        }
        this.push(tag);

    }

    @Override
    public void endElement(String uri, String localName, String name)
            throws SAXException {

        NewLexicalTags tag=NewLexicalTags.valueOf(name.toUpperCase());

        switch (tag) {
            ...
            case TEXT:
                this.mAnnotationListener.notifyAnnotations(...);
        }
        this.pop();

    }

    @Override
    public void characters(char ch[], int start, int length)
            throws SAXException {

        String sentence = new String(ch, start, length);
        switch ((NewLexicalTags)this.get()) {
            ...
        }

    }
}
```

Figure 7.3. An example of XML parser.

SUPWSD: Adding New Modules

We, now, illustrate how to implement new modules for SUPWSD and integrate them into the framework at various stages of the pipeline.

[3] http://liblinear.bwaldvogel.de

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvm

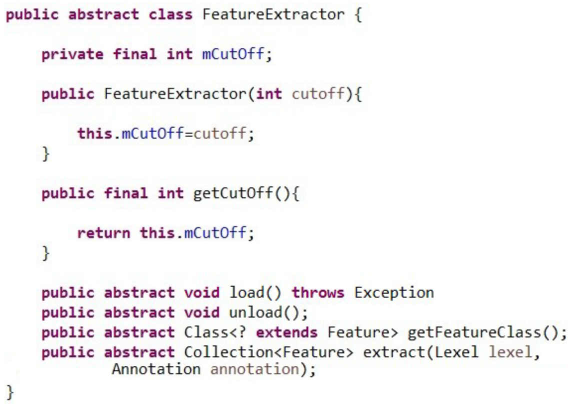
[5] https://wordnet.princeton.edu

[6] http://babelnet.org

[7] The MFS is based on the lexicographic order provided by the sense inventory (either WordNet or BabelNet).

Adding a new input parser. In order to integrate a new XML parser, it is enough to extend the `XMLHandler` class and implement the methods `startElement`, `endElement` and `characters` (see the example in Figure 7.3). With the global variable `mAnnotationListener`, the programmatic user can directly specify when to transmit the parsed text to the text preprocessing module. Instead, in order to integrate a general parser for custom text, it is enough to extend the `Parser` class and implement the `parse` method. An example is provided by the `PlainParser` class that implements a parser for a plain textual file.

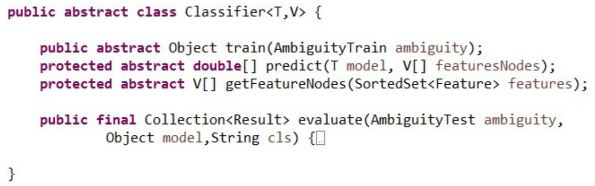
Adding a new preprocessing module. To add a new preprocessing module into the pipeline, it is enough to implement the interfaces in the package `modules.preprocessing.units`. It is also possible to add a brand new step to the pipeline (e.g., a Named Entity Recognition module) by extending the class `Unit` and implementing the methods to load the models asynchronously.



```
public abstract class FeatureExtractor {

    private final int mCutOff;

    public FeatureExtractor(int cutoff){

        this.mCutOff=cutoff;
    }

    public final int getCutOff(){

        return this.mCutOff;
    }

    public abstract void load() throws Exception
    public abstract void unload();
    public abstract Class<? extends Feature> getFeatureClass();
    public abstract Collection<Feature> extract(Lexel lexel,
            Annotation annotation);
}
```

Figure 7.4. The abstract class modeling a feature extractor.

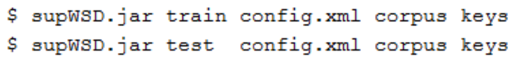
Adding a new feature. A new feature for SUPWSD can be implemented with a two-step procedure. The first step consists in creating a class that extends the abstract class `Feature`. The builder of this class requires a unique key and a name. It is also possible to set a default value for the feature by implementing the method `getDefaultValue`. The second step consists in implementing an extractor for the new feature via the abstract class `FeatureExtractor` (Figure 7.4). Each `FeatureExtractor` has a cut-off value and declares the name of the class through the method `getFeatureClass`.



```
public abstract class Classifier<T,V> {

    public abstract Object train(AmbiguityTrain ambiguity);
    protected abstract double[] predict(T model, V[] featuresNodes);
    protected abstract V[] getFeatureNodes(SortedSet<Feature> features);

    public final Collection<Result> evaluate(AmbiguityTest ambiguity,
            Object model,String cls) {□

}
```

Figure 7.5. The abstract class modeling a classifier.

Adding a new classifier. A new classifier for SUPWSD can be implemented by extending the generic abstract class `Classifier` (Figure 7.5), which declares the methods to train and test the models. Feature conversion is carried out with the generic method `getFeatureNodes`.



```
$ supWSD.jar train config.xml corpus keys
$ supWSD.jar test  config.xml corpus keys
```

Figure 7.6. Command line usage for SUPWSD.

SUPWSD: Usage

SUPWSD can be used effectively via the command line with just 4 parameters (Figure 7.6): the first parameter toggles between the train and test mode; the second parameter contains the path to the configuration file; the third and fourth parameters contain the paths to the dataset and the associated key file (i.e., the file containing the annotated senses for each target word) respectively.

Figure 7.2 shows an example configuration file for SUPWSD. As illustrated, the SUPWSD pipeline is entirely customizable by changing these configuration parameters, and allows the user to employ specific settings at each stage of the pipeline (from preprocessing to actual classification). The `working directory` tag encodes the path in the file system where the trained models are to be saved. Finally, the `writer` tag enables the user to choose the preferred way of printing the test results (e.g., with or without confidence scores for each sense).

SUPWSD can also be used programmatically through its Java API, either using the toolkit (the main class `SupWSD`, provided with the two static methods `train` and `test`, shares the same usage of the command line interface) or using an HTTP RESTful service.

Tr. Corpus	System	Senseval-2	Senseval-3	SemEval-07	SemEval-13	SemEval-15
SemCor	IMS	70.9	69.3	61.3	65.3	69.5
	SupWSD	71.3	68.8	60.2	65.8	70.0
	IMS+emb	71.0	69.3	60.9	**67.3**	71.3
	SupWSD+emb	**72.7**	**70.6**	63.1	66.8	71.8
	IMS$_{-s}$+emb	72.2	70.4	62.6	65.9	71.5
	SupWSD$_{-s}$+emb	72.2	70.3	**63.3**	66.1	71.6
	Context2Vec	71.8	69.1	61.3	65.6	**71.9**
	MFS	65.6	66.0	54.5	63.8	67.1
SemCor + OMSTI	IMS	72.8	69.2	60.0	65.0	69.3
	SupWSD	72.6	68.9	59.6	64.9	69.5
	IMS+emb	70.8	68.9	58.5	66.3	69.7
	SupWSD+emb	**73.8**	**70.8**	**64.2**	**67.2**	71.5
	IMS$_{-s}$+emb	73.3	69.6	61.1	66.7	70.4
	SupWSD$_{-s}$+emb	73.1	70.5	62.2	66.4	70.9
	Context2Vec	72.3	68.2	61.5	**67.2**	**71.7**
	MFS	66.5	60.4	52.3	62.6	64.2

Table 7.1. F-scores (%) of different models in five all-words WSD datasets.

Evaluation and Speed Comparisons

We evaluated SupWSD on the evaluation framework of Chapter 4, which includes five test sets from the Senseval/SemEval series and two training corpus of different size, i.e., SemCor [Miller et al., 1993] and OMSTI [Taghipour and Ng, 2015a]. As sense inventory, we used WordNet 3.0 [Miller, 1995] for all open-class parts of speech. We compared SupWSD with the original implementation of IMS, including the best configurations reported in Iacobacci et al. [2016] which exploit word embedding as features. As shown in Table 7.1, the performance of SupWSD consistently matches up to the original implementation of IMS in terms of F-Measure, sometimes even outperforming its competitor by a considerable margin; this suggests that a neat and flexible implementation not only brings benefits in terms of usability of the software, but also impacts on the accuracy of the model.

We additionally carried out an experimental evaluation on the performance of SupWSD in terms of execution time. As in the previous experiment, we compared SupWSD with IMS and, given that both implementations are written in Java, we tested their programmatic usage within a Java program. We relied on a testing corpus with 1M words and more than 250K target instances to disambiguate, and we used both frameworks on SemCor and OMSTI as training sets. All experiments were performed using an Intel i7-4930K CPU 3.40GHz twelve-core machine. Figures in

	IMS	SUPWSD
train SemCor/sec.	~ 360	~ 120
train SemCor+OMSTI/sec.	~ 3000	~ 510
test/sec.	~ 110	~ 22

Table 7.2. Speed comparison for both the training and testing phases.

Table 7.2 show a considerable gain in execution time achieved by SUPWSD, which is around 3 times faster than IMS on SemCor, and almost 6 times faster than IMS on OMSTI.

Conclusion

In this Appendix we presented SUPWSD, a flexible toolkit for supervised Word Sense Disambiguation which is designed to be modular, highly customizable and easy to both use and extend for end users and researchers. Furthermore, beside the Java API, SUPWSD provides an HTTP RESTful service for programmatic access to the SUPWSD framework and the pre-trained models.

Our experimental evaluation showed that, in addition to its flexibility, SUPWSD can replicate or outperform the state-of-the-art results reported by the best supervised models on standard benchmarks, while at the same time being optimized in terms of execution time.

The SUPWSD framework (including the source code, the pre-trained models, and an online demo) is available at `http://github.com/SI3P/SupWSD`.

Bibliography

Rodrigo Agerri, Josu Bermudez, and German Rigau. Ixa pipeline: Efficient and ready to use multilingual nlp tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3823–3828, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

Eneko Agirre and Philip Edmonds. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media, 2007.

Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics, 2009.

Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. Two graph-based algorithms for state-of-the-art wsd. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 585–593. Association for Computational Linguistics, 2006.

Eneko Agirre, Oier Lopez De Lacalle, Christiane Fellbaum, Andrea Marchetti, Antonio Toral, and Piek Vossen. Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 123–128, 2010.

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

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. Massively Multilingual Word Embeddings. *CoRR*, abs/1602.01925, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, November 2016. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR Workshop*, 2015.

Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

Satanjeev Banerjee and Ted Pedersen. An adapted Lesk algorithm for Word Sense Disambiguation using WordNet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'02, pages 136–145, Mexico City, Mexico, 2002.

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

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43:209–226, 2009.

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, June 2014. Association for Computational Linguistics.

Ander Barrena, Aitor Soroa, and Eneko Agirre. Combining mention context and hyperlinks from wikipedia for named entity disambiguation. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 101–105, 2015.

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

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

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3:1137–1155, 2003.

Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain, April 2017. Association for Computational Linguistics.

Johannes Bjerva, Barbara Plank, and Johan Bos. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

Francis Bond and Ryan Foster. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

Samuel Brody and Mirella Lapata. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111. Association for Computational Linguistics, 2009.

Bruce G Buchanan and David C Wilkins. *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*. Morgan Kaufmann Publishers Inc., 1993.

Thang D Bui, Sujith Ravi, and Vivek Ramavajjala. Neural graph machines: Learning neural networks using graphs. *arXiv preprint arXiv:1703.04818*, 2017.

Iacer Calixto and Qun Liu. Incorporating global visual features into attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1003–1014, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Iacer Calixto, Daniel Stein, Evgeny Matusov, Pintu Lohar, Sheila Castilho, and Andy Way. Using images to improve machine-translating e-commerce product listings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 637–643, Valencia, Spain, April 2017. Association for Computational Linguistics.

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

Jose Camacho-Collados and Roberto Navigli. Babeldomains: Large-scale domain labeling of lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 223–228, Valencia, Spain, 2017. Association for Computational Linguistics.

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

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)*, pages 741–751, Beijing, China, July 2015b. Association for Computational Linguistics.

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.

José Camacho Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017). Vancouver, Canada*, pages 15–26, 2017.

Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *EMNLP-CoNLL 2007-Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61–72, 2007.

Rich Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

Yee Seng Chan and Hwee Tou Ng. Scaling up word sense disambiguation via parallel texts. In *AAAI*, volume 5, pages 1037–1042, 2005.

Devendra Singh Chaplot, Pushpak Bhattacharyya, and Ashwin Paranjape. Unsupervised word sense disambiguation using markov random field and dependency parser. In *AAAI*, pages 2217–2223, 2015.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, 2017a. Association for Computational Linguistics.

Meihao Chen, Zhuoru Lin, and Kyunghyun Cho. Graph convolutional networks for classification with a structured label space. *arXiv preprint arXiv:1710.04908*, 2017b.

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. Association for Computational Linguistics, 2014.

Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

Massimiliano Ciaramita and Yasemin Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 594–602. Association for Computational Linguistics, 2006.

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

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20:37–46, 1960.

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

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, 2008.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, 2007. Association for Computational Linguistics.

Bharath Dandala, Chris Hokamp, Rada Mihalcea, and Razvan C. Bunescu. Sense clustering using Wikipedia. In *Proceedings of Recent Advances in Natural Language Processing*, pages 164–171, Hissar, Bulgaria, 2013.

Jordi Daude, Lluis Padro, and German Rigau. Validation and tuning of wordnet mapping techniques. In *Proceedings of RANLP*, pages 117–123, 2003.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, 2009.

Antonio Di Marco and Roberto Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39: 709–754, 2013.

Mona Diab. Relieving the data acquisition bottleneck in word sense disambiguation. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, pages 303–310. Association for Computational Linguistics, 2004.

S Dolan. Six Degrees of Wikipedia, 2008. URL `http://mu.netsoc.ie/wiki/`.

Philip Edmonds and Scott Cotton. Senseval-2: overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5. Association for Computational Linguistics, 2001.

Steffen Eger, Rüdiger Gleim, and Alexander Mehler. Lemmatization and morphological tagging in german and latin: A comparison and a survey of the state-of-the-art. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1507–1513. European Language Resources Association (ELRA), 2016.

Andreas Eisele and Yu Chen. MultiUN: A Multilingual Corpus from United Nation Documents. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 2868–2872, 2010.

Luis Espinosa Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language*

Processing, pages 424–435, Austin, Texas, 2016. Association for Computational Linguistics.

Luis Espinosa-Anke, Horacio Saggion, Francesco Ronzano, and Roberto Navigli. ExTaSem! Extending, Taxonomizing and Semantifying Domain Terminologies. In *Proceedings of the 30th Conference on Artificial Intelligence (AAAI'16)*, 2016.

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, June 2016. Association for Computational Linguistics.

Stefano Faralli and Roberto Navigli. A new minimally-supervised framework for domain word sense disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1411–1422. Association for Computational Linguistics, 2012.

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, May–June 2015. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California, June 2016. Association for Computational Linguistics.

Erwin Fernandez-Ordonez, Rada Mihalcea, and Samer Hassan. Unsupervised word sense disambiguation with multilingual representations. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 847–851, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Multiwibi:

The multilingual wikipedia bitaxonomy project. *Artificial Intelligence*, 241: 66–102, 2016.

Lucie Flekova and Iryna Gurevych. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041, Berlin, Germany, 2016. Association for Computational Linguistics.

Marc Franco-Salvador, Paolo Rosso, and Manuel Montes-y Gómez. A systematic study of knowledge graph analysis for cross-language plagiarism detection. *Information Processing & Management*, pages 550–570, 2016.

E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. IJCAI, 2007.

E Gabrilovich, M Ringgaard, and A Subramanya. FACC1: Freebase annotation of ClueWeb corpora, Version 1. *Release date*, pages 06–26, 2013.

Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438:900–901, 2005.

Hila Gonen and Yoav Goldberg. Semi supervised preposition-sense disambiguation using multilingual data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2718–2729, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

Aitor González, German Rigau, and Mauro Castillo. A graph-based method to improve Wordnet domains. In *Proceedings of 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, pages 17–28, 2012.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 748–756, 2015.

Alex Graves. Generating Sequences With Recurrent Neural Networks. *CoRR*, abs/1308.0850, 2013.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18: 602–610, 2005.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics.

Weiwei Guo and Mona T. Diab. Combining orthogonal monolingual and multilingual sources of evidence for all words WSD. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1542–1551, Uppsala, Sweden, 2010.

Taher H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, pages 517–526, Hawaii, USA, 2002.

Karl Moritz Hermann and Phil Blunsom. Multilingual Distributed Representations without Word Alignment. In *Proc. of ICLR*, pages 1–9, 2014.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. Learning to understand phrases by embedding the dictionary. *arXiv preprint arXiv:1504.00548*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Eduard 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.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR*, abs/1508.01991, 2015.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensembed: Learning sense embeddings forword and relational similarity. In *53rd Annual*

Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP 2015. Association for Computational Linguistics (ACL), 2015.

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. Association for Computational Linguistics.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

David Jurgens and Roberto Navigli. It's all fun and games until someone annotates: Video games with a purpose for linguistic annotation. *Transactions of the Association for Computational Linguistics*, 2:449–464, 2014.

Mikael Kågebäck and Hans Salomonsson. Word sense disambiguation using a bidirectional lstm. In *5th Workshop on Cognitive Aspects of the Lexicon (CogALex)*. Association for Computational Linguistics, 2016.

Adam Kilgarriff. English lexical sample task description. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 17–20. Association for Computational Linguistics, 2001.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42:21–40, 2008.

Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT summit*, volume 5, pages 79–86, 2005.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the*

14th Conference of the European Chapter of the Association for Computational Linguistics, pages 540–549, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

Pierre Lafon. Sur la variabilité de la fréquence des formes dans un corpus. *Mots*, 1: 127–165, 1980.

Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 41–48. Association for Computational Linguistics, 2002.

Els Lefever and Veronique Hoste. SemEval-2010 Task 3: Cross-lingual Word Sense Disambiguation. In *Proc. of SemEval*, pages 15–20, 2010.

Els Lefever and Véronique Hoste. SemEval-2013 task 10: Cross-lingual Word Sense Disambiguation. In *Proc. of SemEval*, pages 158–166, 2013.

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual Conference on Systems Documentation,* Toronto, Ontario, Canada, pages 24–26, 1986.

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, September 2015. Association for Computational Linguistics.

Antonio Lieto, Enrico Mensa, and Daniele P Radicioni. A resource-driven approach for anchoring linguistic resources to conceptual spaces. In *AI* IA 2016 Advances in Artificial Intelligence*, pages 435–449. Springer, 2016.

Kenneth C Litkowski. Senseval-3 task: Word-sense disambiguation of wordnet glosses. In *In Proc. of SENSEVAL-3 Workshop on Sense Evaluation, in the 42th Annual Meeting of the Association for Computational Linguistics (ACL 2004*, 2004.

Frederick Liu, Han Lu, and Graham Neubig. Handling homographs in neural machine translation. *arXiv preprint arXiv:1708.06510*, 2017.

Oier Lopez de Lacalle and Eneko Agirre. A methodology for word sense disambiguation at 90% based on large-scale crowdsourcing. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 61–70, Denver, Colorado, 2015. Association for Computational Linguistics.

Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multitask sequence to sequence learning. In *ICLR Workshop*, 2016.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics.

Suresh Manandhar, Ioannis P Klapaftis, Dmitriy Dligach, and Sameer S Pradhan. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 63–68. Association for Computational Linguistics, 2010.

Massimiliano Mancini, Jose 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, August 2017. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

Héctor Martínez Alonso and Barbara Plank. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain, April 2017. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53. Association for Computational Linguistics, 2007.

Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 51–61, 2016.

Rada Mihalcea. Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 196–203, Rochester, New York, April 2007. Association for Computational Linguistics.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. The senseval-3 english lexical sample task. In *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text*, 2004.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

Tristan Miller, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych. DKPro WSD: A generalized UIMA-based framework for word sense disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 37–42, 2013.

David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

Andrea Moro and Roberto Navigli. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*, 2015.

Andrea Moro, Roberto Navigli, Francesco Maria Tucci, and Rebecca J. Passonneau. Annotating the masc corpus with babelnet. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4214–4219, 2014a.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244, 2014b.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Roi Reichart, Ira Leviant, Milica Gašić, Anna Korhonen, and Steve Young. Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *TACL*, 5, 2017.

Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:1–69, 2009.

Roberto Navigli. A quick tour of word sense disambiguation, induction and related approaches. In *SOFSEM 2012: Theory and practice of computer science*, pages 115–129. Springer, 2012.

Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250, 2012.

Roberto Navigli and Paola Velardi. Structural Semantic Interconnections: a knowledge-based approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1075–1088, 2005.

Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35. Association for Computational Linguistics, 2007.

Roberto Navigli, David Jurgens, and Daniele Vannella. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of SemEval 2013*, pages 222–231, 2013.

Steven Neale, Luís Gomes, Eneko Agirre, Oier López de Lacalle, and António Branco. Word sense-aware machine translation: Including senses as contextual features for improved translation models. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2777–2783, 2016.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, 2016.

Thanapon Noraset, Chandra Bhagavatula, and Doug Downey. Adding high-precision links to wikipedia. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 651–656, Doha, Qatar, October 2014. Association for Computational Linguistics.

Arantxa Otegi, Nora Aranberri, Antonio Branco, Jan Hajic, Martin Popel, Kiril Ivanov Simov, Eneko Agirre, Petya Osenova, Rita Valadas Pereira, João Ricardo Silva, et al. Qtleap wsd/ned corpora: Semantic annotation of parallel corpora in six languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3023–3030. European Language Resources Association (ELRA), 2016.

Lluís Padró and Evgeny Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2473–2479, Istanbul, Turkey, 2012. European Language Resources Association (ELRA).

Alexander Panchenko, Fide Marten, Eugen Ruppert, Stefano Faralli, Dmitry Ustalov, Simone Paolo Ponzetto, and Chris Biemann. Unsupervised, knowledge-free, and interpretable word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 91–96. Association for Computational Linguistics, 2017a.

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 the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1,* *Long Papers*, pages 86–98, Valencia, Spain, 2017b. Association for Computational Linguistics.

Tommaso Pasini and Roberto Navigli. Train-o-matic: Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 78–88, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

Rebecca J. Passonneau, Collin Baker, Christiane Fellbaum, and Nancy Ide. The MASC Word Sense Sentence Corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3025–3030, 2012.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Tommaso Petrolito and Francis Bond. A Survey of WordNet Annotated Corpora. In *Proceedings of the 7th Global WordNet Conference*, pages 236–245, 2014.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*, 2011.

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.

Mohammad Taher Pilehvar and Roberto Navigli. A large-scale pseudoword-based evaluation framework for state-of-the-art Word Sense Disambiguation. *Computational Linguistics*, 40, 2014.

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*, pages 1341–1351, Sofia, Bulgaria, 2013.

Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. Towards a seamless integration of word senses into downstream nlp applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1857–1869, Vancouver, Canada, 2017. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, August 2016. Association for Computational Linguistics.

Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich Word Sense Disambiguation rivaling supervised system. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1522–1531, Uppsala, Sweden, 2010.

Marten Postma, Ruben Izquierdo, Eneko Agirre, German Rigau, and Piek Vossen. Addressing the mfs bias in wsd systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, may 2016.

Quentin Pradet, Gaël De Chalendar, and Jeanne Baguenier Desormeaux. Wonef, an improved, expanded and evaluated automatic french translation of wordnet. *Proceedings of the 7th Global WordNet Conference, Tartu, Estonia*, pages 32–39, 2014.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*, pages 87–92, 2007.

Alessandro Raganato, Jose Camacho-Collados, Antonio Raganato, and Yunseo Joung. Semantic indexing of multilingual corpora and its application on the history domain. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 140–147, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

Stephen D Richardson, William B Dolan, and Lucy Vanderwende. Mindnet: acquiring and structuring semantic information from text. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 1098–1102. Association for Computational Linguistics, 1998.

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. Association for Computational Linguistics.

Adriana Roventini, Antonietta Alonge, Nicoletta Calzolari, Bernardo Magnini, and Francesca Bertagna. Italwordnet: a large semantic database for italian. In *Language Resources and Evaluation*, 2000.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *New methods in language processing*. Routledge, 2013.

Lenhart Schubert. Turing's dream and the knowledge challenge. In *Proceedings of the national conference on artificial intelligence*, volume 21, pages 1534–1538, 2006.

Walid Shalaby and Wlodek Zadrozny. Measuring semantic relatedness using mined semantic analysis. *arXiv preprint arXiv:1512.03465*, 2015.

Hui Shen, Razvan Bunescu, and Rada Mihalcea. Coarse to Fine Grained Sense Disambiguation in Wikipedia. In *Proc. of SEM*, pages 22–31, 2013.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2012-015*, 2012.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

Benjamin Snyder and Martha Palmer. The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, 2004.

Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 231–235, 2016.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15:1929–1958, 2014.

Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

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.

Kaveh Taghipour and Hwee Tou Ng. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 314–323, Denver, Colorado, 2015b. Association for Computational Linguistics.

Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 442–457. Springer, 2009.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622, 2006.

Rocco Tripodi and Marcello Pelillo. A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, pages 31–70, 2017.

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-Volume 1*, pages 1476–1485. Association for Computational Linguistics, 2011.

Daniele Vannella, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli. Validating and extending semantic knowledge bases using video games with a purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1304. Association for Computational Linguistics, June 2014.

Paola Velardi, Stefano Faralli, and Roberto Navigli. OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction. *Computational Linguistics*, 39: 665–707, 2013.

Noortje Venhuizen, Kilian Evang, Valerio Basile, and Johan Bos. Gamification for word sense labeling. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, 2013.

Oriol Vinyals and Quoc V. Le. A Neural Conversational Model. In *Proc. of ICML, JMLR: W&CP*, volume 37, 2015.

Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781, 2015.

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, June 2016. Association for Computational Linguistics.

Ivan Vulić and Anna Korhonen. On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 247–257, Berlin, Germany, August 2016. Association for Computational Linguistics.

Yogarshi Vyas and Marine Carpuat. Sparse bilingual word representations for cross-lingual lexical entailment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1187–1197, San Diego, California, June 2016. Association for Computational Linguistics.

Wenhui Wang and Baobao Chang. Graph-based dependency parsing with bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany, August 2016. Association for Computational Linguistics.

Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. Multi-objective optimization for the joint disambiguation of nouns and named entities. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 596–605, Beijing, China, 2015. Association for Computational Linguistics.

Robert West, Ashwin Paranjape, and Jure Leskovec. Mining missing hyperlinks from human navigation traces: A case study of wikipedia. In *Proceedings of the 24th international conference on World Wide Web*, pages 1242–1252. International World Wide Web Conferences Steering Committee, 2015.

Fei Wu and Daniel S Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics, 2010.

Z. Wu and C. Lee Giles. Sense-aware semantic analysis: A multi-prototype word representation model using Wikipedia. AAAI, 2015.

Deyi Xiong and Min Zhang. A sense-based translation model for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1459–1469, Baltimore, Maryland, 2014. Association for Computational Linguistics.

Jay Young, Valerio Basile, Lars Kunze, Elena Cabrio, and Nick Hawes. Towards lifelong object learning by integrating situated robot perception and semantic web mining. In *Proceedings of the European Conference on Artificial Intelligence conference*, pages 1458–1466, The Hague, Netherland, 2016.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, 2016.

Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701, 2012.

Zhi Zhong and Hwee Tou Ng. Word sense disambiguation for all words without hard labor. In *IJCAI*, pages 1616–1622, 2009.

Zhi Zhong and Hwee Tou Ng. It Makes Sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proceedings of the ACL System Demonstrations*, pages 78–83, 2010.

Zhi Zhong and Hwee Tou Ng. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 273–282. Association for Computational Linguistics, 2012.

Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. 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 1127–1137, Beijing, China, July 2015. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics.

Geoffrey Zweig and Christopher JC Burges. The microsoft research sentence completion challenge. Technical report, Technical Report MSR-TR-2011-129, Microsoft, 2011.