Domain Adaptation and Training Data Acquisition in Wide-Coverage Word Sense Disambiguation and its Application to Information Retrieval

Zhong Zhi

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the School of Computing

NATIONAL UNIVERSITY OF SINGAPORE

2012

Abstract

Word Sense Disambiguation (WSD) is the process of identifying the meaning of an ambiguous word in context. It is considered a fundamental task in Natural Language Processing (NLP).

Previous research shows that supervised approaches achieve state-of-the-art accuracy for WSD. However, the performance of the supervised approaches is affected by several factors, such as domain mismatch and the lack of sense-annotated training examples. As an intermediate component, WSD has the potential of benefiting many other NLP tasks, such as machine translation and information retrieval (IR). But few WSD systems are integrated as a component of other applications.

We release an open source supervised WSD system, IMS (It Makes Sense). In the evaluation on lexical-sample tasks of several languages and English all-words tasks of SensEval workshops, IMS achieves state-of-the-art results. It provides a flexible platform to integrate various feature types and different machine learning methods, and can be used as an all-words WSD component with good performance for other applications.

To address the domain adaptation problem in WSD, we apply the feature augmentation technique to WSD. By further combining the feature augmentation technique with active learning, we greatly reduce the annotation effort required when adapting a WSD system to a new domain.

One bottleneck of supervised WSD systems is the lack of sense-annotated

training examples. We propose an approach to extract sense annotated examples from parallel corpora without extra human efforts. Our evaluation shows that the incorporation of the extracted examples achieves better results than just using the manually annotated examples.

Previous research arrives at conflicting conclusions on whether WSD systems can improve information retrieval performance. We propose a novel method to estimate the sense distribution of words in short queries. Together with the senses predicted for words in documents, we propose a novel approach to incorporate word senses into the language modeling approach to IR and also exploit the integration of synonym relations. Our experimental results on standard *TREC* collections show that using the word senses tagged by our supervised WSD system, we obtain statistically significant improvements over a state-of-the-art IR system.

Contents

List of Figures **v**

List of Tables **vii**

Chapter 1 Introduction **1**

1.1 Approaches for Word Sense Disambiguation 2

1.2 Knowledge Resources for Word Sense Disambiguation 3

1.3 SensEval Workshops 5

1.4 Difficulties in Supervised Word Sense Disambiguation 8

1.5 Applications of Word Sense Disambiguation 9

1.6 Contributions of This Thesis 10

1.6.1 A High Performance Open Source Word Sense Disambiguation System 11

1.6.2 Domain Adaptation for Word Sense Disambiguation 11

1.6.3 Automatic Extraction of Training Data from Parallel Corpora 12

1.6.4 Word Sense Disambiguation for Information Retrieval 12

1.7 Organization of This Thesis 12

Chapter 2 Related Work **14**

2.1 Knowledge Based Approaches 14

2.2 Supervised Learning Approaches 16

2.2.1 Word Sense Disambiguation as a Classification Problem . . . 17
2.2.2 Tackling the Bottleneck of Lack of Training Data 18
2.2.3 Domain Adaptation for Word Sense Disambiguation 20
2.3 Semi-supervised Learning Approaches 21
2.4 Unsupervised Learning Approaches 23
2.5 Applications of Word Sense Disambiguation 23
2.5.1 Word Sense Disambiguation in Statistical Machine Translation 24
2.5.2 Word Sense Disambiguation in Information Retrieval 26
2.5.3 Word Sense Disambiguation in Other NLP Tasks 28

Chapter 3 An Open Source Word Sense Disambiguation System 30
3.1 System description 31
3.1.1 System Architecture 32
3.1.1.1 Preprocessing 32
3.1.1.2 Feature and Instance Extraction 33
3.1.1.3 Classification 35
3.1.2 The Training Data Set for English All-Words Tasks 35
3.2 Experiments 37
3.2.1 Lexical-Sample Tasks 37
3.2.1.1 English Lexical-Sample Tasks 37
3.2.1.2 Lexical-Sample Tasks of Other Languages 38
3.2.2 English All-Words Tasks 41
3.3 Summary 42

Chapter 4 Domain Adaptation for Word Sense Disambiguation 44
4.1 Experimental Setting 45
4.2 In-Domain and Out-of-Domain Evaluation 47
4.2.1 Training and Evaluating on OntoNotes 47

4.2.2 Using Out-of-Domain Training Data 49
4.3 Concatenating In-Domain and Out-of-Domain Data for Training . . 49
4.3.1 The Feature Augmentation Technique for Domain Adaptation 50
4.3.2 Experiments 51
4.4 Active Learning for Domain Adaptation 53
4.4.1 Active learning with the Feature Augmentation Technique for Domain Adaptation 54
4.4.2 Experiments 56
4.5 Summary 58

Chapter 5 Automatic Extraction of Training Data from Parallel Corpora 59

5.1 Acquiring Training Data from Parallel Corpora 60
5.2 Automatic Selection of Chinese Translations 62
5.2.1 Academia Sinica Bilingual Ontological WordNet 63
5.2.2 A Common English-Chinese Bilingual Dictionary 63
5.2.3 Shortening Chinese Translations 65
5.2.4 Using Word Similarity Measure 66
5.2.4.1 Calculating Chinese Word Similarity 67
5.2.4.2 Assigning Chinese Translations to English Senses Based on Word Similarity 68
5.3 Evaluation 70
5.3.1 Quality of the Automatically Selected Chinese Translations . 70
5.3.2 Experiments on OntoNotes 71
5.4 Summary 74

Chapter 6 Word Sense Disambiguation for Information Retrieval 75

6.1 The Language Modeling Approach to IR 77

6.1.1 The Language Modeling Approach 77
6.1.2 Pseudo Relevance Feedback 78
6.1.2.1 Collection Enrichment 80
6.2 Word Sense Disambiguation 80
6.2.1 Word Sense Disambiguation System 80
6.2.2 Estimating Sense Distributions for Query Terms 82
6.3 Incorporating Senses into Language Modeling Approaches 84
6.3.1 Incorporating Senses 84
6.3.2 Expanding with Synonym Relations 86
6.4 Experiments 88
6.4.1 Experimental Settings 88
6.4.2 Experimental Results 91
6.5 Summary 96

Chapter 7 Conclusion **97**
7.1 Future Work 98

List of Figures

3.1 IMS system architecture 31

4.1 WSD accuracies evaluated on section 23, with different sections as training data. 48

4.2 WSD accuracies evaluated on section 23, using SemCor and different OntoNotes sections as training data. ON: only OntoNotes as training data. SC+ON: SemCor and OntoNotes as training data, SC+ON Augment: Concatenating SemCor and OntoNotes via the Augment domain adaptation technique. 52

4.3 The active learning algorithm. 55

4.4 Results of applying active learning with the feature augmentation technique on different number of word types. Each curve represents the adaptation process of applying active learning on a certain number of most frequently occurring word types. 57

5.1 Assigning Chinese translations to English senses using word similarity measure. 69

5.2 Significance test results on all noun types. 74

6.1 The process of generating senses for query terms 83

List of Tables

1.1 SensEval-2 results 6
1.2 SensEval-3 results 6
1.3 SemEval-2007 results 7

3.1 Statistics of the word types which have training data for WordNet-1.7.1 sense-inventory. 36
3.2 Statistics of English lexical-sample tasks 38
3.3 WSD accuracies on SensEval English lexical-sample tasks 38
3.4 Statistics of SensEval-3 Italian, Spanish, and Chinese lexical-sample tasks 39
3.5 WSD accuracies on SensEval-3 Italian, Spanish, and Chinese lexical-sample tasks 40
3.6 WSD accuracies on SensEval/SemEval fine-grained and coarse-grained all-words tasks 41

4.1 Size of the sense-annotated data in the various WSJ sections. 46

5.1 Senses of the noun "article" in WordNet 61
5.2 Size of English-Chinese parallel corpora 62
5.3 Statistics of sense-annotated nouns in OntoNotes 2.0 71
5.4 WSD accuracy on OntoNotes 2.0 72

5.5 Error reduction comparing to SC baseline 73

6.1 Statistics of query sets 89

6.2 Results on the test sets in MAP score. The first three rows show the results of the top participating systems, the next row shows the performance of the baseline method, and the remaining rows are the results of our method with different settings. Single dagger (†) and double dagger (‡) indicate statistically significant improvement over $Stem_{prf}$ at the 95% and 99% confidence level with a two-tailed paired t-test, respectively. The best results are highlighted in bold. 92

Acknowledgments

This thesis is the result of six years of work during which I have been accompanied and supported by many people. It is now my great pleasure to take this opportunity to thank them.

First and foremost, I would like to express my sincerest gratitude and deepest respect to my supervisor Prof. Ng Hwee Tou for his continuous support during the whole period of my Ph.D study. Prof. Ng not only provided me insightful feedback and ideas, but also taught me the meaning of rigorous research. Without his guidance, expertise, patience, and understanding, the completion of this thesis would not have been possible.

I sincerely thank Prof. Tan Chew Lim and Prof. Sim Khe Chai for serving on my doctoral committee. Their constructive comments at various stages have been significantly useful in shaping the thesis up to completion.

I also want to thank many of my present and past colleagues from the Computational Linguistics lab: Chan Yee Seng, Qiu Long, Zhao Shanheng, Chia Tee Kiah, Hendra Setiawan, Lu Wei, Zhao Jin, Lin Ziheng, Wang Pidong, Daniel Dahlmeier, Na Seung-Hoon, Zhu Muhua, Zhang Hui, *etc.* Special thanks to Chan Yee Seng for his great help at the early stage of my graduate study, Qiu Long for proof-reading my thesis, and all the colleagues for sharing the joy and pain of my Ph.D journey.

I am grateful to my friends in Singapore: Lu Huanhuan, Wang Xianjun, Wang Xiangyu, Zeng Zhiping, Zhang Dongxiang, and Zhuo Shaojie. They have given me a lot of help and encouragement in my research as well as my daily life. We had a wonderful time together and I will definitely miss it.

Last but not least, I would like to thank my family, especially my parents, for their support and understanding.

To my parents, Gong Daolin and Zhong Yuezhu.

Chapter 1

Introduction

In natural languages, many words have multiple meanings. For example, in the following two sentences:

"He works in a bank as a cashier."

"We took a walk along the river bank."

the two occurrences of the word *bank* denote two different meanings: financial institution and sloping land, respectively. The particular meaning of an ambiguous word can be determined by its context. A word sense is a representation of one meaning of a word. The task of identifying the correct sense of an ambiguous word in context is known as word sense disambiguation (WSD).

As a basic semantic understanding task at the lexical level, WSD is a fundamental problem in natural language processing (NLP), and is considered as an intermediate and essential task of many other NLP tasks. For example, in machine translation, resolving the sense ambiguity is a necessity to correctly translate an ambiguous word. In the field of information retrieval, the ambiguity of query and document terms can affect the retrieval performance. In addition, WSD has the potential of benefiting other NLP tasks which require a certain degree of semantic interpretation, such as text classification, sentiment analysis, *etc.*

1.1 Approaches for Word Sense Disambiguation

WSD has been investigated for decades (Ide and Veronis, 1998; Agirre and Edmonds, 2006). In the early years, researchers tried to build rule-based systems using hand crafted knowledge sources to disambiguate word senses. However, because hand-written rules can only be developed by linguistic experts and each word needs its own rules, creating rule-based systems incurs extremely high cost.

With the development of large amounts of machine readable resources and machine learning methods, researchers turned to automatic methods for WSD. These automatic methods can be categorized into four types:

- **Knowledge based approaches** Knowledge based WSD approaches utilize the definitions or some other knowledge sources given in machine readable dictionaries or thesauruses. The performance of systems using these approaches greatly relies on the availability of knowledge sources.

- **Supervised approaches** Supervised approaches treat WSD as a classification problem. They employ machine learning methods to train classifiers from a set of sense-annotated data, and then the appropriate senses are predicted as the class labels of the target ambiguous words by the trained classifiers. The performance of supervised WSD methods is dependent on the size of the sense-annotated training data.

- **Semi-supervised approaches** Semi-supervised WSD approaches use a small amount of sense-annotated data together with a large amount of unannotated raw data to train better classifiers. However, the performance of semi-supervised WSD methods is unstable.

- **Unsupervised approaches** Unsupervised WSD approaches do not use any manually annotated resources. Senses are induced from a large amount

of unannotated raw corpora, and WSD is viewed as a clustering problem. The drawback of unsupervised methods is that the real meaning of each individual word cannot be ascertained after clustering without human annotation.

Two baseline methods are widely used for WSD, *the random baseline* and *the most frequent sense (MFS) baseline*. The former randomly selects one of all possible senses with equal probabilities. Usually, it is considered as the lower bound of WSD. Different from the random baseline, the MFS baseline always picks the most frequent sense in a corpus for each word occurrence. It achieves better performance than the random baseline and many knowledge-based approaches.

1.2 Knowledge Resources for Word Sense Disambiguation

Machine readable dictionaries or thesauri, such as *the Collins English Dictionary*, *the Longman Dictionary of Contemporary English*, *the Omega Ontology*, *the Oxford Dictionary of English*, and *WordNet*, are important knowledge resources for NLP. These dictionaries provide the sense inventories for WSD. The knowledge resources in these dictionaries, such as sense definitions and semantic relations, are also widely used by WSD systems.

Among these dictionaries and thesauri, WordNet (Miller, 1995) is the most commonly used one for WSD. WordNet[1] is a lexical database of English developed at Princeton University. It provides senses for content words, i.e., nouns, verbs, adjectives and adverbs. In WordNet, senses with the same meaning are grouped into a synonym set, called a *synset*. Besides the gloss and several examples which illustrate the usage for each synset, WordNet also provides various semantic relations which link different synsets, such as hypernymy/hyponymy, holonymy/meronymy,

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

and so on. Both nouns and verbs in WordNet are organized into hierarchies, defined by the hypernymy/hyponymy relation. At the top level, WordNet has 25 primitive groups of nouns and 15 groups of verbs. Because the senses for each word are sorted by decreasing frequency based on one part of the Brown Corpus, known as SemCor (Miller et al., 1994), the first sense of each word in WordNet (WNs1) is usually considered as the most frequent sense in a general domain. Thus WNs1 can be considered as the MFS baseline in a general domain. With the success of WordNet in English, WordNets in several other languages have been developed, such as the WordNet Libre du Francais[2](WOLF) for French, MultiWordNet[3] for Italian, the Academia Sinica Bilingual Ontological WordNet[4](BOW) for Chinese, FinnWordNet[5] for Finnish, and EuroWordNet[6] for several European languages.

Another important kind of resources for WSD is the sense-annotated corpora. Here we list several widely used sense-annotated corpora:

- The SemCor corpus (Miller et al., 1994) is one of the most widely used publicly available sense-annotated corpora created by Princeton University. As a subset of the Brown Corpus, SemCor contains more than 230,000 manually tagged content words with WordNet senses. Current supervised WSD systems usually rely on this relatively small corpus for training examples.

- The DSO corpus was developed at the Defense Science Organization (DSO) of Singapore (Ng and Lee, 1996). It consists of about 190,000 word occurrences of 191 word types from the Brown corpus and Wall Street Journal corpus with WordNet senses.

- The Open Mind Word Expert (OMWE) project (Chklovski and Mihalcea,

[2]http://alpage.inria.fr/~sagot/wolf.html
[3]http://multiwordnet.fbk.eu/english/home.php
[4]http://bow.sinica.edu.tw/
[5]http://www.ling.helsinki.fi/en/lt/research/finnwordnet/
[6]http://www.illc.uva.nl/EuroWordNet/

2002) is another sense-annotated corpus with WordNet senses, which were annotated by Internet users. This data set is used in the SensEval-3 English lexical sample task.

- OntoNotes (Hovy et al., 2006) is a sense-annotated corpus created more recently. It is a project which aimed to annotate a large corpus with several layers of semantic annotations, including coreference, word senses, *etc.*, for three languages (Arabic, Chinese, and English). For its WSD part, OntoNotes groups fine-grained WordNet senses into coarse-grained senses and forms a coarse-grained sense inventory. It manually annotates senses for instances of nouns and verbs with inter-annotator agreement (ITA) of 90%, based on a coarse grained sense inventory.

1.3 SensEval Workshops

Before SensEval, there exist few common data sets publicly available for testing WSD systems. Therefore, it was difficult to compare the performance of WSD systems. SensEval[7] is an international evaluation exercise devoted to the evaluation of WSD systems. It aims to test the strengths and weaknesses of WSD systems on different words in various languages.

After the first SensEval workshop SensEval-1 in 1998, SensEval-2 was held in 2001, SensEval-3 in 2004, SemEval-2007 in 2007, and SemEval-2010 in 2010. They provided considerable test data covering many languages, including English, Arabic, Chinese, Spanish, etc. The data sets of SensEval workshops are considered the standard benchmark data sets for evaluating WSD systems.

SensEval workshops have two classic WSD tasks, lexical-sample task and all-words task. In the lexical-sample task, participants are required to label a set

[7]http::/www.sensevels.org

lexical-sample		all-words	
System	Accuracy	System	Accuracy
JHU (R)	64.2%	SMUaw	69.0%
SMUls	63.8%	CNTS-Antwerp	63.6%
KUNLP	62.9%	Sinequa-LIA-HMM	61.8%
MFS	47.6%	WNs1	62.4%

Table 1.1: SensEval-2 results

of target words in the test data set. Training data with the manually sense tagged target words in context is provided for each target word in this task. In contrast, no training data is provided in the all-words task. Participants are allowed to use any external resources to label all the content words in a text.

lexical-sample		all-words	
System	Accuracy	System	Accuracy
htsa3	72.9%	GAMBL-AW	65.2%
IRST-Kernels	72.6%	SenseLearner	64.6%
nusels	72.4%	Koc University	64.1%
MFS	55.2%	WNs1	62.4%

Table 1.2: SensEval-3 results

Both SensEval-2 and SensEval-3 had the English lexical sample task and the English all-words task. SensEval-2 used WordNet-1.7 as the sense inventory, and SensEval-3 used WordNet-1.7.1 as the sense inventory. Table 1.1 and Table 1.2 present the results of the top participating systems and the MFS/WNs1 baseline in SensEval-2 and SensEval-3, respectively (Kilgarriff, 2001; Palmer et al., 2001; Mihalcea, Chklovski, and Kilgarriff, 2004; Snyder and Palmer, 2004). The WNs1 baseline method achieves relatively high performance on the English all-words tasks. Most of the top systems are supervised and they outperform the systems using the other methods including the MFS/WNs1 baseline. However, the accuracies of these top systems are only around 70% or lower. In fact, the inter annotator/tagger agreement (ITA) reported for manual sense-tagging on these

SensEval English lexical-sample and English all-words datasets is typically in the mid-70s. For example the ITA is only 67.3% in SensEval-3 lexical-sample task (Mihalcea, Chklovski, and Kilgarriff, 2004) and 72.5% in SensEval-3 English all-words task (Snyder and Palmer, 2004). Therefore, the poor performance of WSD systems can be attributed to the fine granularity of the sense inventory of WordNet. Using a fine-grained sense inventory is considered as one of the obstacles to effective WSD.

coarse-grained lexical-sample		fine-grained all-words		coarse-grained all-words	
System	Accuracy	System	Accuracy	System	Accuracy
NUS-ML	88.7%	PNNL	59.1%	NUS-PT	82.5%
UBC-ALM	86.9%	NUS-PT	58.7%	NUS-ML	81.6%
I2R	86.4%	UNT-Yahoo	58.3%	LCC-WSD	81.5%
MFS	78.0%	MFS	51.4%	MFS	78.9%

Table 1.3: SemEval-2007 results

Therefore, in SemEval-2007, besides a fine-grained English all-words task using WordNet-2.1 as the sense inventory, a coarse-grained English all-words task and a coarse-grained English lexical-sample task were organized (Navigli, Litkowski, and Hargraves, 2007; Pradhan et al., 2007). The coarse-grained English lexical-sample task used the coarse-grained sense inventory of OntoNotes, and the coarse-grained English all-words task used a sense inventory which has the WordNet senses mapped to the Oxford Dictionary of English to form a relatively coarse-grained sense inventory. The top participating WSD systems achieve more than 80% accuracy in the two coarse-grained tasks. It proves that sense granularity has an important impact on the accuracy figures of current state-of-the-art WSD systems.

1.4 Difficulties in Supervised Word Sense Disambiguation

The results of the SensEval workshops show that supervised WSD approaches are better than the other approaches and achieve the best performance. However, the performance of supervised WSD systems is constrained by several factors.

The first problem is the granularity of the sense inventory. As presented in the last section, for the English tasks in the SensEval workshops, which used WordNet as the sense inventory, the WSD accuracies of the top systems were only around 70%. The accuracies of WSD systems improved to over 80% in the coarse-grained English tasks of SemEval-2007. The improvement in these coarse-grained tasks shows that an appropriate sense granularity is important for a WSD system to achieve high accuracy.

Similar to other NLP tasks which rely on supervised learning algorithms, supervised WSD systems also suffer from the problem of lack of sense-annotated training examples. Comparing the performance of the top WSD systems in the English lexical-sample tasks and the English all-words tasks in SensEval workshops, we observe that the accuracies in the English lexical-sample tasks are higher than those in the English all-words tasks. One reason is that a large amount of training data were provided for the target word types in lexical-sample tasks, but it is hard to gather such large quantities of training data for all word types. The sense annotation process is laborious and time-consuming, such that very few sense-annotated corpora are publicly available. SemCor has just 10 instances for each word type on average, which is too small to train a supervised WSD system for English. Considering the vocabulary size of English, supervised WSD methods faces the word coverage problem in the all-words task. Therefore, it is important to reduce the human efforts needed in annotating new training examples as well as

scaling up the coverage of sense-annotated corpora.

Another problem faced by supervised WSD approaches is the domain adaptation problem. The need for domain adaptation is a general and important issue for many NLP tasks (Daumé III and Marcu, 2006). For instance, semantic role labeling (SRL) systems are usually trained and evaluated on data drawn from WSJ. In the CoNLL-2005 shared task on SRL (Carreras and Màrquez, 2005), however, a task of training and evaluating systems on different domains was included. For that task, systems that were trained on the PropBank corpus (Palmer, Gildea, and Kingsbury, 2005) (which was gathered from WSJ) suffered a 10% drop in accuracy when evaluated on test data drawn from the Brown Corpus, compared to the performance achievable when evaluated on data drawn from WSJ. More recently, CoNLL-2007 included a shared task on dependency parsing (Nivre et al., 2007). In this task, systems that were trained on Penn Treebank (drawn from WSJ) but evaluated on data drawn from a different domain (such as chemical abstracts and parent-child dialogues) showed a similar drop in performance. For research involving training and evaluating WSD systems on data drawn from different domains, several prior research efforts (Escudero, Màrquez, and Riagu, 2000; Martinez and Agirre, 2000) observed a similar drop in performance of about 10% when a WSD system that was trained on the Brown Corpus part of the DSO corpus was evaluated on the WSJ part of the corpus, and vice versa. Similar to the problem of lack of training data, it is hard to annotate a large corpus for every new domain because of the expenses of manual sense annotation. Thus, domain adaptation is essential for the application of supervised WSD systems across different domains.

1.5 Applications of Word Sense Disambiguation

Besides the study of WSD as an isolated problem, its applications in other tasks have also been investigated.

The need for WSD in machine translation (MT) was first pointed out by Weaver (1955). WSD system is expected to help select proper translations for MT systems. However, some attempts show that WSD can hurt the performance of MT systems (Carpuat and Wu, 2005). More recently, researchers demonstrate that WSD can improve the performance of state-of-the-art MT systems by using the target translation phrases as the senses (Chan, Ng, and Chiang, 2007; Carpuat and Wu, 2007; Giménez and Màrquez, 2007). This shows that the appropriate integration of WSD is important to its applications in other tasks.

WSD is necessary for information retrieval (IR) to resolve the ambiguity of query words. Similar to its application in MT, different attempts show conflicting conclusions. Some researchers reported a drop in retrieval performance by using word senses (Krovetz and Croft, 1992; Voorhees, 1993). Some other experiments observed improvements by integrating word senses in IR systems (Schütze and Pedersen, 1995; Gonzalo et al., 1998; Stokoe, Oakes, and Tait, 2003; Kim, Seo, and Rim, 2004). Therefore, it is still not clear whether a WSD system can improve the performance of IR.

Besides MT and IR, WSD has also been attempted in other high-level NLP tasks such as text classification, sentiment analysis, *etc.* The ultimate goal of WSD is to benefit these tasks in which WSD is needed. However, there are a limited number of successful applications of WSD. Prior work often reported conflicting results on whether WSD is helpful for some NLP tasks. Therefore, more work is needed to evaluate the utility of WSD in NLP applications.

1.6 Contributions of This Thesis

In this thesis, we tackle some of the difficulties listed in Section 1.4 and apply WSD to improve the performance of IR. The contributions of this thesis are as follows.

1.6.1 A High Performance Open Source Word Sense Disambiguation System

To promote WSD and its applications, we build an English all-words supervised WSD system, IMS (It Makes Sense) (Zhong and Ng, 2010). As an open source WSD toolkit, the extensible and flexible platform of IMS allows researchers to try out various preprocessing tools, WSD features, as well as different machine learning algorithms. IMS functions as a high performance WSD system. We also provide classifier models for English trained with the sense-annotated examples collected from parallel texts, SemCor, and the DSO corpus. Therefore, researchers who are not interested in WSD can directly use IMS as a WSD component in other tasks. Evaluation on several SensEval English lexical-sample tasks shows that IMS is a start-of-the-art WSD system. IMS also achieve high performance in the evaluation on SensEval English all-words tasks. It shows that the classifier models for English in IMS are of high quality and have a wide coverage of English words.

1.6.2 Domain Adaptation for Word Sense Disambiguation

Domain adaptation is a serious problem for supervised learning algorithms. In (Zhong, Ng, and Chan, 2008), we employed the feature augmentation technique to address this problem in WSD. In our experiment, we used the Brown Corpus as the source domain and the Wall Street Journal corpus as the target domain. The results show that the feature augmentation technique can significantly improve the performance of WSD in the target domain, given small amount of target domain training data. We further proposed a method of incorporating the feature augmentation technique into the active learning process to acquire training examples for a new domain. This method greatly reduced the human efforts required in sense-annotating the words in a new domain.

1.6.3 Automatic Extraction of Training Data from Parallel Corpora

To tackle the bottleneck of lack of sense-annotated training data of WSD, in (Zhong and Ng, 2009), we extended the work of Ng *et al.*(2003) and Chan and Ng (2005a) to gather training examples from parallel texts. Instead of using human annotated Chinese translations, we proposed a completely automatic approach to gather Chinese translations. Our approach relies on English-Chinese parallel corpora, English-Chinese bilingual dictionaries, and automatic methods of finding synonyms of Chinese words. With our approach, in the process of extracting sense annotated data from parallel texts, no additional human sense annotation or word translation is needed. Thus it can easily scale up WSD to all words in English.

1.6.4 Word Sense Disambiguation for Information Retrieval

The language modeling approach with pseudo relevance feedback is one of the best IR approaches. In (Zhong and Ng, 2012) , we successfully integrated word senses into the language modeling approach to improve the performance of IR. We proposed a novel model to incorporate senses into the language modeling approach and further explored the incorporation of synonym relations into our model. In the evaluation on several TREC tasks, our system outperformed the language modeling IR approach and achieved very competitive performance compared to the TREC participating systems.

1.7 Organization of This Thesis

The remainder of this thesis is organized as follows. Chapter 2 introduces the related work of WSD. We describe our open source supervised WSD system and present its evaluation on several test data sets in Chapter 3. In Chapter 4, we

apply the feature augmentation technique to address the domain adaption problem of WSD. We further integrate the feature augmentation technique into the active learning algorithm to improve the annotation efficiency of the training data for a new domain. In Chapter 5, we describe our method of extracting training data from parallel texts without expensive human effort and evaluate the quality of the gathered training data on OntoNotes senses. In Chapter 6, we apply WSD to the IR task. We modify the language modeling IR approach and achieve significant improvement on several TREC tasks. Finally, we conclude in Chapter 7.

Chapter 2

Related Work

In this chapter, we briefly review the WSD approaches and the applications of WSD in other tasks. Further details of the background literature in the field can be found in (Agirre and Edmonds, 2006). We will introduce knowledge based approaches, supervised learning approaches, semi-supervised learning approaches, and unsupervised learning approaches. Then, we will discuss the applications of WSD in machine translation, information retrieval, and other NLP tasks.

2.1 Knowledge Based Approaches

Knowledge based WSD approaches rely on external knowledge sources to identify the word senses. They make use of definitions and semantic relations in machine readable dictionaries or thesauri.

The Lesk Algorithm (Lesk, 1986) is the first well-known WSD method based on machine readable dictionaries. It identifies senses of ambiguous words by counting word overlaps between the dictionary definitions of each word in the surrounding context. The sense that leads to the highest overlap is selected for each word. Kilgarriff and Rosenzweig (2000) introduced a simpler Lesk Algorithm, which only

counts overlaps between the dictionary definition of the target word sense and the bag of words in context. Comparing to the original Lesk Algorithm, the simpler version is more straightforward, but it is reported to be better than the original Lesk Algorithm (Vasilescu, Langlais, and Lapalme, 2004).

Because dictionary definitions are usually short, the Lesk Algorithm does not work well. Lesk (1986) suggested that the example sentences in a dictionary can be considered as part of the dictionary definition. Moreover, many variants of the Lesk Algorithm have been proposed to improve its performance (Vasilescu, Langlais, and Lapalme, 2004). Instead of using a standard dictionary, some methods utilize a thesaurus like WordNet which provides a rich hierarchy of semantic relations to disambiguate word sense. Banerjee and Pedersen (2002) extended dictionary definitions by considering the synsets that are related to the word senses in WordNet. Besides word overlap, various semantic similarity measures are used to calculate the connectivities between the senses of a sequence of words (Rada et al., 1989; Lin, 1997; Jiang and Conrath, 1997; Resnik, 1999; Pedersen, Patwardhan, and Michelizzi, 2004). The senses with the maximum relatedness with the content words in the surrounding context are picked for each ambiguous word. The WordNet::Similarity package provides several different measures of relatedness of word senses with the semantic relations and sense definitions in WordNet (Pedersen, Patwardhan, and Michelizzi, 2004). In (Pedersen, Banerjee, and Patwardhan, 2005), they evaluated the usage of these semantic similarity measures in WSD and concluded that the extended gloss overlap measure is the most effective.

Another kind of knowledge based approach is graph-based approaches. This kind of approach exploits the graph structures of a sequence of words to perform disambiguation. To come up with a graph representation, the senses of each word in a text are represented as the vertices in a graph. Two vertices are connected with an edge if they have some semantic relation. These semantic relations can be ex-

tracted from WordNet, sense-annotated corpora, or dictionaries of collocations. In (Mihalcea, Tarau, and Figa, 2004; Mihalcea, 2005), the PageRank algorithm (Brin and Page, 1998) was applied to pick the sense with the highest rank for each word as the answer. In (Sinha and Mihalcea, 2007), they extended their previous work by using a collection of semantic similarity measures and graph-based centrality algorithms. Navigli and Velardi (2005) proposed the Structural Semantic Interconnections (SSI) algorithm which selects the senses with the maximal connectivity degree in the graph. Navigli and Lapata (2007) studied different graph-based centrality algorithms for deciding the relevance of vertices with the semantic relations in WordNet. In (Navigli and Lapata, 2010) and (Ponzetto and Navigli, 2010), they extended their previous work by enriching WordNet relations and achieved improvement. Agirre and Soroa (2007) exploited the relation types in a lexical knowledge base, Multilingual Central Repository. They found that all the relations in the lexical knowledge base are valuable and the relations coming from the sense-annotated corpora are the most influential. In (Agirre and Soroa, 2009), they extended their previous work by using Personalized PageRank (Jeh and Widom, 2003) and concluded that the Personalized PageRank outperforms the traditional PageRank.

Knowledge based approaches do not depend on high quality sense-annotated corpora. With the development of large-scale machine readable knowledge sources, these approaches have wide coverage of words. In general, the performance of knowledge based approaches is not as good as supervised approaches.

2.2 Supervised Learning Approaches

Supervised learning approaches tackle the WSD problem by using machine learning methods to train classifiers from sense-annotated corpora. As highlighted in Section

1.3, supervised WSD systems outperform the other WSD approaches and achieve the best performance in SensEval workshops (Kilgarriff, 2001; Palmer et al., 2001; Snyder and Palmer, 2004; Mihalcea, Chklovski, and Kilgarriff, 2004; Pradhan et al., 2007; Navigli, Litkowski, and Hargraves, 2007). However, the performance of supervised WSD approaches greatly relies on the amount of available high quality sense-annotated corpora. Because manual sense annotation is expensive, the size of sense-annotated corpora becomes the bottleneck of supervised learning approaches.

In this section, we first review different supervised learning approaches for WSD and the features they used. Then, we review several approaches which try to tackle the bottleneck of lack of sense-annotated training data. Finally, we review the domain adaptation problem in WSD.

2.2.1 Word Sense Disambiguation as a Classification Problem

Supervised learning approaches treat WSD as a classification problem. In supervised learning approaches, machine learning approaches are employed to train a classifier for each ambiguous word with sense-annotated corpora and the features extracted from them. The classifier assigns the most probable sense out of a set of predefined senses as the class label to each occurrence of the target word.

Many types of knowledge sources are used as features for the supervised learning systems, such as surrounding words in context, local collocations, parts-of-speech (POS) of neighboring words, syntactic relations, semantic class information, and subjectivity information (Yarowsky, 1994; Ng and Lee, 1996; Ng, 1997a; Lee and Ng, 2002; Dang and Palmer, 2005; Wiebe and Mihalcea, 2006). Generally, a combination of knowledge sources gives better performance than using a single knowledge source (Lee and Ng, 2002).

Using the knowledge sources mentioned above as features, various super-

vised learning methods have been applied to WSD. Yarowsky (1994; 2000) used decision lists to disambiguate a word by measuring collocational distribution in log likelihoods. Ng and Lee (1996) and Veenstra *et al.* (2000) employed exemplar-based approaches to assign each test instance with the label of its nearest training instance by measuring the distances between each test instance and the training instances. In addition, several well-known classification methods, such as Naïve Bayes (NB), Support Vector Machines (SVM), Maximum Entropy (ME), and Decision Trees (DT) also achieve good performance in WSD (Pedersen, 2000; Lee and Ng, 2002; Tratz et al., 2007). In one comparison of different supervised learning methods, SVM achieves the best performance (Lee and Ng, 2002). It also achieves state-of-the-art performance on several evaluations in SensEval workshops.

Because different classifiers have different biases and strengths, in many works (Pedersen, 2000; Klein et al., 2002; Florian et al., 2002), researchers attempted to combine different classifiers with various combination methods, such as count-based and probability-based voting, confidence-based combination, performance-based combination, and meta-voting. Their experiments showed that the combined system obtains a significantly lower error rate compared to the individual classifiers.

2.2.2 Tackling the Bottleneck of Lack of Training Data

Although supervised learning approaches achieve great success in WSD, as highlighted in Section 1.4, their performance is greatly affected by the availability of sense-annotated training examples. In the past decades, researchers have devoted great efforts to manual sense-annotation on text corpora. However, as shown in Section 1.2, the size of available sense-annotated corpora is still insufficient to train a high-accuracy WSD system for all words of English. Many researchers attempt to solve this problem by using the existing sense-annotated corpora as much as possible or reducing the human effort of annotating new corpora. However, the lack

of sense-annotated training examples is still a challenging problem for supervised learning approaches to WSD.

To tackle this bottleneck, some researchers attempt to use the existing training data of one word as the training data for other words. Kohomban and Lee (2005) tried to use training examples of words different from the actual word to be classified, by exploiting WordNet semantic relations. Each synset in WordNet is a descendant of some unique beginner. To disambiguate a target word, they trained coarse-grained classifiers for the unique beginners with the training instances of the words which have the same unique beginner as the target word using TiMBL, a memory based method. Using some heuristic, they mapped the classification result on unique beginners into finer grained senses as the answer. They reported competitive performance in the evaluation on SensEval English all-words tasks. Ando (2006) applied the Alternating Structure Optimization (ASO) algorithm to WSD. ASO is a machine learning method for learning predictive structure shared by multiple prediction problems via joint empirical risk minimization. With ASO, the sense disambiguation process of one ambiguous word could benefit from the training data of other words. The evaluation on SensEval lexical sample tasks shows that the ASO algorithm obtained consistent improvement across several languages and tasks.

Active learning is another promising way to solve the lack of sense-annotated training data (Ng, 1997b; Fujii et al., 1998; Chklovski and Mihalcea, 2002; Chen et al., 2006; Chan and Ng, 2007; Zhu and Hovy, 2007). In each iteration of active learning, classifiers select the most informative unlabeled instance for humans to annotate. In this way, the human labeling effort becomes most effective. Zhu and Hovy (2007) introduced an active learning algorithm with resampling for WSD. The resampling techniques they used include under-sampling, over-sampling, or bootstrap-based over-sampling (an over-sampling method based on the bootstrap

technique).

Multilingual resources are also used in WSD to automatically acquire sense-annotated training instances, based on the observation that the translations of the different senses of an ambiguous word are typically be different in a second language (Resnik and Yarowsky, 1997; Diab and Resnik, 2002; Ng, Wang, and Chan, 2003; Chan and Ng, 2005a). In (Ng, Wang, and Chan, 2003; Chan and Ng, 2005a), English-Chinese parallel texts were exploited for WSD. Chinese translations were manually assigned for each sense of a target English word beforehand. The sense of an English word in a word aligned English-Chinese parallel corpus is identified by the Chinese translation that the English word is aligned to. Compared to sense-annotating training examples directly, the human effort needed in the approach of (Chan and Ng, 2005a) is drastically reduced. The system NUS-PT built using this approach (Chan, Ng, and Zhong, 2007) was the best performing system in the coarse-grained English all-words task in SemEval-2007.

As parallel corpora are not widely available for all language pairs, Wang and Carroll (2005) extended Chan and Ng's work with the help of bilingual dictionaries and large quantities of texts of another language. They first used an English-Chinese dictionary to translate the senses of an English word into Chinese words, and then retrieved text snippets that contained these Chinese words from a large Chinese corpus. Next, the Chinese snippets were translated back to English using a Chinese-English dictionary. These English translations were regarded as the sense examples for each sense. However, their experiment showed that the quality of the instances generated by their method was far behind that of (Chan and Ng, 2005a).

2.2.3 Domain Adaptation for Word Sense Disambiguation

The domain adaptation problem is commonly encountered in supervised learning methods. This problem limits the performance of supervised WSD systems. In

the experiments of Escudero *et al.* (2000), classifiers trained in one domain were found to have an inferior performance when applied to another domain. Generally speaking, the performance of a WSD system trained on data from one domain will drop when applied on texts from a different domain.

To tackle the domain adaptation problem in WSD, one can either make use of domain adaptation techniques or retrain a WSD system with some extra domain-specific sense annotated training data.

Because sense distribution tend to be different across domains, McCarthy *et al.* (2004) proposed a method to predict the predominant sense or the most frequent sense in a corpus. When the predominant sense of a word in a test corpus is different from the training corpus, using the predicted predominant sense in the test corpus and relying on the most frequent sense heuristic gives a respectable baseline performance.

Instead of predicting the predominant sense, Chan and Ng (2005b) proposed a method to estimate the sense distribution in a new domain. They used naïve Bayes as the supervised learning algorithm to provide posterior probabilities in a target domain corpus. In (Chan and Ng, 2006), they improved their method by using well calibrated probabilities to estimate the sense priors more accurately.

Besides different sense distributions, the classification clues may also vary in different domains. In (Chan and Ng, 2007), they applied active learning to domain adaptation for WSD. They combined predicted predominant sense information and count merging in the process of active learning, and greatly reduced the human effort needed in the adaptation process.

2.3 Semi-supervised Learning Approaches

Different from supervised learning approaches, semi-supervised learning approaches only require a small amount of sense-annotated training data as seeds to generate

more sense-annotated instances from raw corpora. In this way, the supervised learning approaches can have a larger set of sense-annotated training data.

Hearst (1991) presented a bootstrapping WSD system for the disambiguation of noun homographs using large text corpora. In each iteration, the system automatically acquires additional statistical information from instances newly disambiguated with certainty. Different from using large text corpora, Mihalcea (2002) made use of the Web as a big corpus. Her system queried Web search engines with the seeds generated from existing training data. The instances from Web documents were disambiguated and added to the set of seeds and the generation process continued.

In another work, Mihalcea (2004) investigated the application of co-training to the bootstrapping process for WSD. In this system, two or more classifiers were trained and each classifier independently selected new labeled instances to add to the original set of training instances. Pham *et al.* (2005) investigated the use of unlabeled training data with four semi-supervised learning methods: co-training, smoothed co-training, spectral graph transduction, and spectral graph transduction with co-training. Their experimental results on SensEval-2 English lexical-sample task and all-words task show that unlabeled data can bring improvement in WSD accuracy and spectral graph transduction with co-training outperforms the other three methods as well as a naïve Bayes baseline.

Niu *et al.* (2005) performed WSD using a semi-supervised learning approach with label propagation. In label propagation, each instance is represented as a vertex in an edge weighted connected graph. The information of vertices corresponding to labeled instances in the graph is propagated to connected vertices through the weighted edges until the graph achieves a globally stable state. Each unlabeled instance will be assigned a tag according to the label information in its corresponding vertex.

2.4 Unsupervised Learning Approaches

Unsupervised learning approaches are often referred to as "Word Sense Discrimination" or "Word Sense Induction". These approaches treat WSD as a clustering problem and they do not use any external knowledge sources or sense-annotated corpora (Schütze, 1992; Schütze, 1998).

In unsupervised learning approaches, the occurrences of ambiguous words are clustered based on the similarity of contexts. Because no dictionary or sense-annotated corpus is used, the sense labels assigned by these approaches are different from the pre-defined senses in dictionaries. Therefore, they cannot be easily evaluated on standard WSD datasets and compared with the other methods. Consequently, in SemEval 2007 and SemEval 2010, word sense induction tasks were defined to allow comparison of word sense induction and discrimination systems (Agirre and Soroa, 2007; Manandhar et al., 2010). Senses can be manually assigned to each cluster predicted by unsupervised WSD systems. In this way, unsupervised learning approaches can reduce the amount of manual sense annotation needed.

2.5 Applications of Word Sense Disambiguation

Regarded as an intermediate task, WSD has been incorporated into the applications of many other NLP tasks. In this section, we review attempts of incorporating WSD to improve the performance of other NLP tasks.

2.5.1 Word Sense Disambiguation in Statistical Machine Translation

Translations in a target foreign language can be different for different senses of a word in a source language. Thus, integrating an accurate WSD system into a Statistical Machine Translation (SMT) system is expected to be helpful for selecting the correct translations for ambiguous words. Although lexical selection has already been done in SMT systems, not as many knowledge sources are used in SMT as in WSD. As a result, lexical selection in SMT is not accurate. Phrase-based SMT systems partly solve this problem by taking advantage of local collocation information in phrases. But similar to words, phrases can also be ambiguous. Therefore, incorporating a WSD system may achieve further improvement in SMT performance.

In previous research, various authors come to conflicting conclusions on whether WSD has any positive impact on SMT. In a pilot study, Brown *et al.* (1991) proposed a method to use a WSD system in a French-English SMT system. In their experiment, positive results are observed. However, their experiment is limited by the simple WSD system they used and the unrealistic assumption that each of the hundreds of words they studied has exactly 2 senses.

Carpuat and Wu (2005) integrated a state-of-the-art Chinese WSD system (Carpuat, Su, and Wu, 2004) in a word-based Chinese-English SMT system to help choose better English translations. In their experiment, HowNet is used as the sense-inventory for Chinese words. The SMT system is forced to use the English translation of the predicted sense output by the WSD system. They reported that WSD system was helpful for very few lexical selections in their experiment, and concluded that WSD hurt the performance of SMT.

In contrast to (Carpuat and Wu, 2005), the translations in the target language in (Vickrey et al., 2005; Cabezas and Resnik, 2005) are used as the senses of each single word in the source language. However, Vickrey *et al.* (2005) just showed

improvement on word translations but not on the complete MT task. Cabezas and Resnik (2005) only achieved a small improvement in BLEU score with no statistical significance tests reported.

Chan *et al.* (2007) integrated a Chinese WSD system in a hierarchical phrase-based SMT system, Hiero. They built WSD classifiers for Chinese phrases consisting of at most 2 Chinese words. The senses of each Chinese phrase are the English words or phrases which are aligned to the Chinese phrase in parallel texts. The output of their WSD system is directly integrated into the tuning and decoding procedures to optimize the translation result. In their experiment, statistically significant improvement in BLEU score is achieved.

Carpuat and Wu (2007) also obtained positive results with integrating a Chinese WSD system into a phrased-based SMT system, Pharaoh. In their work, every Chinese phrase in a given SMT input sentence is disambiguated, with no limitation of the phrase length. Their evaluation on 8 commonly used automated MT metrics showed stable improvements with WSD incorporated. This conclusion is the exact opposite of that in (Carpuat and Wu, 2005). The authors explained that WSD predictions for longer phrases are important to improve translation quality.

Giménez and Màrquez (2007) employed WSD to predict possible phrase translations based on local context in Spanish-to-English MT. In their experiments, their method of predicting phrase translations with WSD techniques outperforms the most frequent translation baseline. However, when they integrated the predicted phrase translations into a phrased-based SMT system, Pharaoh, the BLEU metric did not reflect this improvement. Manual evaluation showed that their method only had gain in adequacy but not fluency. Therefore, they argued that the integration of predicted probabilities into SMT requries further study.

Instead of using the output of a WSD system, Chiang *et al.* (2009) directly integrated WSD-like features such as local collocations into a hierarchical and a

syntax-based MT system. Together with some other target and source side features, both systems achieved significant improvement in BLEU score in their experiment.

According to the above results, SMT systems can benefit from either WSD features or the output of WSD systems. Which of these two alternatives is a better way to integrate WSD in MT is still not clear.

2.5.2 Word Sense Disambiguation in Information Retrieval

The application of WSD in IR has been studied for many years. Many previous studies have analyzed the benefits as well as the problems of applying WSD to IR.

Krovetz and Croft (1992) studied the sense matching between terms in query and the document collection. They concluded that the benefits of WSD in IR are not as expected because query words have skewed sense distribution and the collocation effect from other query terms already performs some disambiguation.

Sanderson (1994; 2000) used pseudowords to introduce artificial word ambiguity in order to study the impact of sense ambiguity on IR. He concluded that because the effectiveness of WSD can be negated by inaccurate WSD performance, high accuracy of WSD is an essential requirement to achieve improvement.

In another work, Gonzalo *et al.* (1998) used a manually sense annotated corpus, SemCor, to study the effects of incorrect disambiguation. They obtained significant improvements by representing documents and queries with accurate senses as well as synsets. Their experiment also showed that with the synset representation, which included synonym information, WSD with an error rate of 40%–50% can still improve IR performance. Their later work (Gonzalo, Penas, and Verdejo, 1999) verified that part of speech information is discriminatory for IR purposes.

Several works attempted to disambiguate terms in both queries and documents with the senses predefined in hand-crafted sense inventories, and then used the senses to perform indexing and retrieval. Voorhees (1993) used the hyper-

nymy/hyponymy relation in WordNet to disambiguate the polysemous nouns in a text. In her experiments, the performance of sense-based retrieval is worse than stem-based retrieval on all test collections. Her analysis showed that inaccurate WSD caused the poor results.

Stokoe *et al.* (2003) employed a WSD system using WordNet as the sense inventory with an accuracy of 62.1% to disambiguate terms in both the text collections and the queries in their experiments. Their evaluation on TREC collections achieved significant improvements over a standard term based vector space model. However, it is hard to judge the effect of word sense disambiguation in their study because of the overall poor performances of their baseline method and their system.

Instead of using a fine-grained sense inventory, Kim *et al.* (2004) tagged words with 25 root senses of nouns in WordNet. Their retrieval method maintained the stem-based index and adjusted the term weight in a document according to its sense matching result with the query. They attributed the improvement achieved on TREC collections to their coarse-grained, consistent, and flexible sense tagging method. The integration of senses into the traditional stem-based index overcomes some of the negative impact of disambiguation errors.

Different from using predefined sense inventories, Schütze and Pedersen (1995) induced the sense inventory directly from the text retrieval collection. For each word, its occurrences were clustered into senses based on the similarity of their contexts. Their experiments showed that using senses improved retrieval performance, and the combination of word-based ranking and sense-based ranking can further improve performance. However, the clustering process of words is a time consuming task. Because the resulting sense inventory is collection dependent, it is also hard to expand the text collection.

Many studies investigated the expansion of the queries to match more documents by using knowledge sources from thesauri. Some researchers achieved im-

provements by expanding the disambiguated query words with synonyms and some other information from WordNet (Voorhees, 1994; Liu et al., 2004; Liu, Yu, and Meng, 2005; Fang, 2008). The usage of knowledge sources from WordNet in document expansion also showed improvements in IR systems (Cao, Nie, and Bai, 2005; Agirre, Arregi, and Otegi, 2010).

2.5.3 Word Sense Disambiguation in Other NLP Tasks

In text categorization, a document is usually represented as a bag of words. Kehagias *et al.* (2003) showed that given gold standard WSD annotations, text categorization accuracy can be improved by 1-2%. But the authors suspected that the errors of automatic WSD systems will offset this improvement. In (Bloehdorn and Hotho, 2004), the authors extended text categorization features with semantic information in WordNet. They achieved statistically significant improvement on three different datasets. However, they only tried the most frequent sense and a simple WSD method, but not a state-of-the-art WSD system.

In subjectivity analysis, Wiebe and Mihalcea (2006) found that word senses can be tagged as subjective or objective, and many words have both subjective and objective senses. Their experiments showed that subjectivity can benefit WSD. In their follow up work (Akkaya, Wiebe, and Mihalcea, 2009), they introduced a variant task of WSD, subjectivity WSD, in which each word occurrence can be tagged as either subjective or objective. They showed that subjectivity WSD can improve the performance of contextual subjectivity and sentiment analysis. In another work, Balamurali, Joshi, and Bhattacharyya (2011) used WordNet senses in supervised sentiment analysis. Their experiments show that using gold standard WSD annotations as features can significantly improve over word-based features. However, the improvement by using automatic WSD is modest.

WSD is needed by many NLP tasks. However, the utility of WSD has been in doubt because of the contrasting conclusions. Previous successful applications show that the method of integrating WSD can be quite different depending on its application.

Chapter 3

An Open Source Word Sense Disambiguation System

As shown in Section 1.3, WSD systems based on supervised learning methods achieved the best performance in SensEval and SemEval workshops. However, there are few publicly available open source WSD systems – the only other publicly available WSD system that we are aware of is SenseLearner (Mihalcea and Csomai, 2005). Therefore, for applications which require WSD as a component, researchers can only make use of some baselines or simple knowledge based methods. This limits the use of WSD in other applications, especially for researchers whose research interests are not in WSD. An open source supervised WSD system will promote the use of WSD in other applications.

In this chapter, we present such a system IMS[1] (It Makes Sense), a supervised learning system for WSD. IMS is implemented in Java, and it provides an extensible and flexible platform for researchers interested in WSD. Different tools can be chosen to perform preprocessing, such as trying out various features in the feature extraction step, and applying different machine learning methods or toolkits in

[1] http://nlp.comp.nus.edu.sg/software/ims

the classification step. By default, we use linear support vector machines as the classifier with multiple features. We also provide classification models for a set of English words. Therefore, IMS can also be an English all-words WSD component for other NLP tasks. Our implementation achieves state-of-the-art results on several SensEval and SemEval tasks.

The next section describes the system architecture and the training data we prepared for English all-words tasks. Then we evaluate our system on SensEval/SemEval lexical-sample and all-words tasks.

3.1 System description

In this section, we outline the IMS system, and introduce the default preprocessing tools, the feature types, and the machine learning methods used in our implementation. Then we briefly explain the collection of training data for English content words.

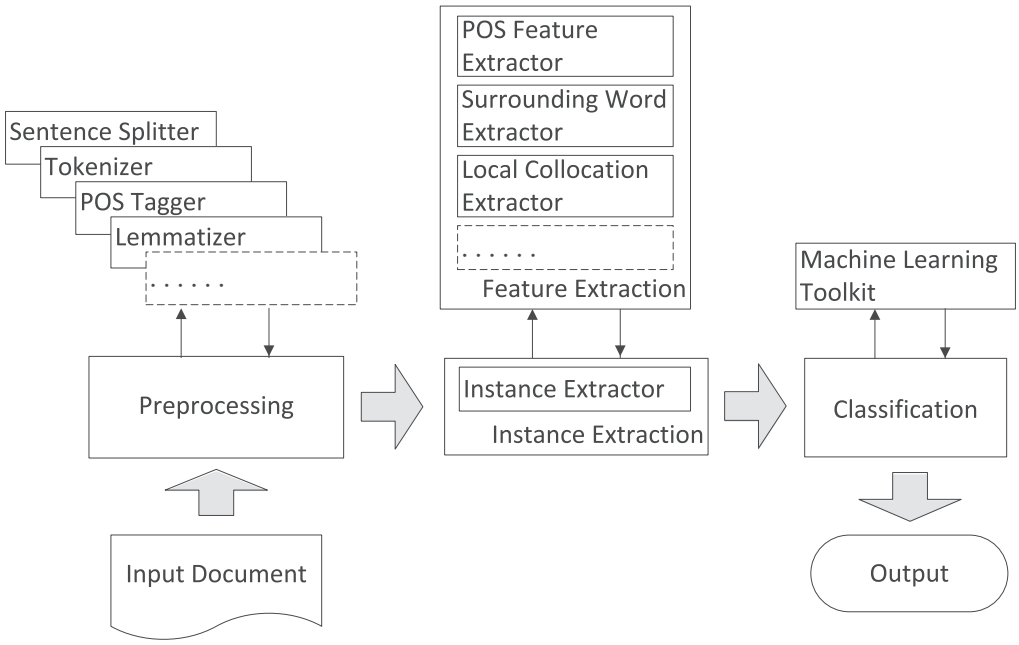

Figure 3.1: IMS system architecture

3.1.1 System Architecture

Figure 3.1 shows the system architecture of IMS. It consists of three independent modules: preprocessing, feature and instance extraction, and classification. Knowledge sources are generated from input texts in the preprocessing step. With these knowledge sources, instances together with their features are extracted in the instance and feature extraction step. Then we train one classification model for each word type.

3.1.1.1 Preprocessing

Preprocessing is the step to convert input texts into formatted information. Users can integrate different tools in this step. These tools are applied on the input texts to extract knowledge sources such as sentence boundaries, part-of-speech (POS) tags, *etc.* The extracted knowledge sources are stored for use in the later steps.

In IMS, default preprocessing is carried out in the following four steps:

- Detecting the sentence boundaries in a raw input text with a *sentence splitter*.
- Breaking a sentence into tokens with a *tokenizer*.
- Assigning POS tags to all tokens with a *POS tagger*.
- Finding the lemma form of each token with a *lemmatizer*.

By default, the sentence splitter and POS tagger in the OpenNLP toolkit[2] are used for sentence splitting and POS tagging; a Java version of Penn Treebank tokenizer[3] is applied in tokenization; and JWNL[4], a Java API for accessing the WordNet thesaurus, is used to find the lemma form of each token.

[2]http://opennlp.sourceforge.net/
[3]http://www.cis.upenn.edu/~treebank/tokenizer.sed
[4]http://jwordnet.sourceforge.net/

3.1.1.2 Feature and Instance Extraction

After gathering the formatted information in the preprocessing step, we use an instance extractor together with a list of feature extractors to generate the instances and their associated features.

Previous research has found that combining multiple knowledge sources achieves high WSD accuracy (Ng and Lee, 1996; Ng, 1997a; Lee and Ng, 2002; Decadt, Hoste, and Daelemans, 2004). We follow the work of (Lee and Ng, 2002) and combine three knowledge sources for all content word types[5]:

- *POS Tags of Surrounding Words*

  We use the POS tags of three words to the left and three words to the right of the target ambiguous word, and the target word itself. The POS tag feature cannot cross sentence boundary, which means all the associated surrounding words should be in the same sentence as the target word. If a word crosses sentence boundary, the corresponding POS tag value will be assigned as *null*.

  For example, suppose we want to disambiguate the word *interest* in a POS-tagged sentence "My/PRP$ brother/NN has/VBZ always/RB taken/VBN a/DT keen/JJ interest/NN in/IN my/PRP$ work/NN ./.". The 7 POS tag features for this instance are $<$*VBN*, *DT*, *JJ*, *NN*, *IN*, *PRP$*, *NN*$>$.

- *Surrounding Words*

  Surrounding word features include all the individual words in the surrounding context of an ambiguous word w. The surrounding words can be in the current sentence or immediately adjacent sentences.

  After stop words and words without alphabetic characters (punctuation symbols and numbers) are removed, the remaining words are converted to their lemma forms in lower case. Each lemma is considered as one binary feature.

[5]Syntactic relations are omitted for efficiency reason.

The feature value is set to be 1 if the corresponding lemma occurs in the surrounding context of w, 0 otherwise.

For example, suppose there is a set of surrounding word features {*account*, *economy*, *rate*, *take*} in the training data set of the word *interest*. For a test instance of *interest* in the sentence "My brother has always taken a keen interest in my work .", the surrounding word feature vector will be <0, 0, 0, 1>.

- *Local Collocations*

  We use 11 local collocation features including:

  Unigram collocations $C_{-2,-2}$, $C_{-1,-1}$, $C_{1,1}$, $C_{2,2}$,

  Bigram collocations $C_{-2,-1}$, $C_{-1,1}$, $C_{1,2}$,

  Trigram collocations $C_{-3,-1}$, $C_{-2,1}$, $C_{-1,2}$, and $C_{1,3}$,

  where $C_{i,j}$ refers to an ordered sequence of words in the same sentence of w. Offsets i and j denote the starting and ending positions of the sequence relative to w, where a negative (positive) offset refers to a word to the left (right) of w.

  For example, suppose in the training data set, the word *interest* has a set of local collocations { *"account ."*, *"of all"*, *"in my"*, *"to be"*} for $C_{1,2}$. For a test instance of *interest* in the sentence "My brother has always taken a keen interest in my work .", the value of feature $C_{1,2}$ will be *"in my"*.

As shown in Figure 3.1, we implement one feature extractor for each feature type. New features can be included by implementing the corresponding feature extractors for them.

3.1.1.3 Classification

In IMS, the classifier trains a model for each word type which has training data. The instances collected in the previous step are converted to the format expected by the machine learning toolkit in use. Thus, the classification step is decoupled from the feature extraction step.

IMS provides module interfaces to *LIBLINEAR*[6] (Fan et al., 2008), *LIBSVM*[7], *MaxEnt*[8], and the *WEKA* machine learning toolkit (Witten and Frank, 2005). We use LIBLINEAR as the default classifier of IMS, with a linear kernel and all the parameters set to their default values.

The trained classification models will be applied to the test instances of the corresponding word types in the testing step. If a test instance word type is not seen during training, we will output its predefined default sense, i.e., the WordNet first sense, as the answer. Furthermore, if a word type has neither training data nor predefined default sense, we will output "U", which stands for the missing/unknown sense, as the answer.

3.1.2 The Training Data Set for English All-Words Tasks

Apart from a supervised WSD system, for the users who only need WSD as a component in their applications, it is also important to provide them the classification models. The performance of a supervised WSD system greatly depends on the size of the sense-annotated training data used. To overcome the lack of sense-annotated training examples, besides the training instances from the widely used sense-annotated corpus SemCor and the DSO corpus, we also follow the approach described in Chan and Ng (2005a) to extract more training examples from parallel texts. We use 6 English-Chinese parallel corpora: Hong Kong Hansards, Hong

[6]http://www.bwaldvogel.de/liblinear-java/
[7]http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[8]http://maxent.sourceforge.net/

Kong News, Hong Kong Laws, Sinorama, Xinhua News, and the English translation of Chinese Treebank. These parallel corpora were already aligned at the sentence level. After tokenizing the English texts and performing word segmentation (Low, Ng, and Guo, 2005) on the Chinese texts, the GIZA++ software (Och and Ney, 2000) is used to perform word alignment on the parallel texts. For each English word e, we have a list of Chinese translations manually assigned to the WordNet senses of e. From the word alignment output of GIZA++, the occurrences of an English word e which are aligned to one of the manually assigned Chinese translations c are selected. Since we know the sense s associated with a Chinese translation c, occurrences of the word e in the English side of the parallel corpora that are aligned to c will be assigned the sense s. The 3 surrounding sentences of these occurrences are extracted as sense-annotated training data for e.

We only extract training examples from parallel texts for the top 60% most frequently occurring polysemous content words in Brown Corpus, which includes 730 nouns, 190 verbs, and 326 adjectives. For each of the top 60% nouns and adjectives, we gather a maximum of 1,000 training examples from parallel texts. For each of the top 60% verbs, we extract not more than 500 examples from parallel texts, as well as up to 500 examples from the DSO corpus. We also make use of the sense-annotated examples from SemCor as part of our training data for all nouns, verbs, adjectives, and 28 most frequently occurring adverbs in Brown Corpus. The experiments of a different version of IMS, NUS-PT (Chan, Ng, and Zhong, 2007), on the SemEval-2007 English all-words task show that adding examples from parallel texts results in accuracy improvements.

POS	noun	verb	adj	adv
# of types	11,445	4,705	5,129	28

Table 3.1: Statistics of the word types which have training data for WordNet-1.7.1 sense-inventory.

Table 3.1 summarizes the number of word types for which we have collected training instances based on WordNet sense inventory version 1.7.1. We generated classification models with the IMS system for over 21,000 word types which we have training data. On average, each word type has 38 training instances. The total size of the models is about 200MB. These models are packaged with IMS for direct use by end users.

3.2 Experiments

In our experiments, we evaluate our IMS system on SensEval and SemEval tasks, the benchmark data sets for WSD. As introduced in Section 1.3, the SensEval workshops have two types of classic WSD tasks: lexical-sample task and all-words task. The evaluation on SensEval and SemEval lexical-sample tasks is to measure the strength of our IMS system, while the evaluation on SensEval and SemEval all-words tasks is to measure the quality of the training data we have collected.

3.2.1 Lexical-Sample Tasks

In SensEval lexical-sample tasks, both the training and the test data sets are provided for each target word tested. We evaluate IMS on the SensEval lexical-sample tasks with three different machine learning toolkits: SVM in LIBLINEAR, SVM in WEKA, and MaxEnt. We evaluate on several English lexical-sample tasks, as well as the Italian, Spanish, and Chinese lexical-sample tasks.

3.2.1.1 English Lexical-Sample Tasks

We apply IMS on SensEval-2 and SensEval-3 English lexical-sample task with its default setting. Table 3.2 lists the statistics of the data sets in these tasks. The first two columns give the number of word types and the average number of senses per

Task	word types	avg. #senses	training instances	test instances
SensEval-2	73	11.5	8,611	4,328
SensEval-3	57	5.4	7,860	3,944

Table 3.2: Statistics of English lexical-sample tasks

word type, respectively. The last two columns show that total number of training instances and test instances, respectively.

Task	SensEval-2	SensEval-3
IMS (LIBLINEAR)	**65.3%**	72.6%
IMS (WEKA)	65.0%	72.0%
IMS (MaxEnt)	62.2%	69.4%
Rank 1	64.2%	**72.9%**
Rank 2	63.8%	72.6%
Rank 3	62.9%	72.4%
MFS	47.6%	55.2%

Table 3.3: WSD accuracies on SensEval English lexical-sample tasks

Table 3.3 compares the performance of our system to the top three systems that participated in the above tasks (Yarowsky et al., 2001; Mihalcea and Moldovan, 2001; Mihalcea, Chklovski, and Kilgarriff, 2004) and the MFS (most frequent sense) baseline. Evaluation results show that IMS achieves significantly better accuracies than the MFS baseline. The accuracies of using linear kernel SVM in WEKA and LIBLINEAR as classifier is 3% higher than using MaxEnt across both tasks. Comparing to the top participating systems, IMS achieves comparable results in both tasks.

3.2.1.2 Lexical-Sample Tasks of Other Languages

Besides the English lexical-sample tasks, we also apply IMS to the lexical-sample tasks of some other languages. We choose three lexical-sample tasks in SensEval-3: Italian, Spanish, and Chinese. Table 3.4 lists some statistics of the data sets

of these three tasks. The first column gives the number of word types, the second column gives the average number of senses per word type, and the last two columns show the total number of training instances and test instances, respectively.

Task	word types	avg. senses	training instances	test instances
Italian	45	6.1	5,145	2,439
Spanish	46	3.3	8,430	4,195
Chinese	20	4.0	793	379

Table 3.4: Statistics of SensEval-3 Italian, Spanish, and Chinese lexical-sample tasks

The Italian lexical-sample task contains 25 nouns, 10 verbs, and 10 adjectives. The number of senses per word type is 6.1. The training set is twice as large as the test set. On average, each word type has 114 training instances and 54 test instances. The sentences in both the training and test data set were tokenized, lemmatized, and POS tagged by the task organizers. With the lemma and POS tag provided for each word, we use the same default setting of IMS for English in this task, with POS tag features, surrounding word features, and local collocation features.

The Spanish lexical-sample task contains 21 nouns, 18 verbs, and 7 adjectives. On average, each word type has 183 training instances and 91 test instances with 3.3 senses per word type. Similar to the Italian lexical-sample task, preprocessing steps including tokenization, lemmatization, and POS tagging were applied on the data set. Therefore, we can also apply the default setting of IMS on the Spanish lexical-sample task.

Different from the English lexical-sample task, the senses of a Chinese word in the Chinese lexical-sample task are not defined with respect to its POS. Therefore, the training data and test data of a Chinese word may contain a mixture of words with different POS. On average, each word has 40 training instances and 19 test instances. POS tags are provided for all words in both training and test data by

the task organizers. The feature setting we use for Chinese is different from the other three languages. We use only three POS tag features: P_{-1}, P_0, and P_1. For the local collocation features, we use 3 features: C_{-1}, C_1, and $C_{-1,1}$. We perform feature selection on surrounding word features by keeping those words which appear 3 or more times in some sense of an ambiguous Chinese word in the training data. We also use a list of Chinese stop words, (with 507 Chinese words or punctuations) to filter the surrounding word features.

	Italian	Spanish	Chinese
IMS (LIBLINEAR)	56.9%	**87.3**%	62.3%
IMS (WEKA)	**57.1**%	87.2%	**63.3**%
IMS (MaxEnt)	56.6%	84.1%	62.5%
Rank 1	53.1%	84.2%	60.4%
Rank 2	51.5%	84.0%	-
Rank 3	49.8%	82.5%	-
MFS	18.3%	67.7%	28.5%

Table 3.5: WSD accuracies on SensEval-3 Italian, Spanish, and Chinese lexical-sample tasks

Table 3.5 lists the performance of our system, the top three participating systems in the lexical-sample tasks (Magnini, Giampiccolo, and Vallin, 2004; Màrquez et al., 2004; Niu, Ji, and Tan, 2004), and the MFS baseline. Similar to the English lexical tasks, IMS easily beats the MFS baseline on all three language. It also achieves better performance than the top participating systems. The evaluation results show that IMS is robust and it performs well on different languages.

The performance of IMS varies across the different languages. The overall performance on Spanish is better than the other two languages. One possible reason is that the Spanish lexical-sample task has more training data and fewer senses per word than the other two tasks. In the Italian lexical-sample task, each word type has 6 senses. The high ambiguity of words may be the cause of the poorer results in this task. Although Chinese has only 4 senses per word, the size of its training

data is relatively small. Thus, the accuracy in the Chinese lexical-sample task is still much worse than Spanish.

3.2.2 English All-Words Tasks

In SensEval or SemEval English all-words tasks, no training data are provided. Thus we choose WNs1 as the baseline, which always selects the first sense in WordNet as the answer for each word.

	SensEval-2	SensEval-3	SemEval-2007	
	Fine-grained	Fine-grained	Fine-grained	Coarse-grained
IMS (LIBLINEAR)	68.2%	**67.6%**	58.3%	**82.6%**
IMS (WEKA)	67.8%	67.5%	**59.1%**	82.2%
IMS (MaxEnt)	67.5%	67.4%	58.9%	82.0%
Rank 1	**69.0%**	65.2%	**59.1%**	82.5%
Rank 2	63.6%	64.6%	58.7%	81.6%
Rank 3	61.8%	64.1%	58.3%	81.5%
WNs1	61.9%	62.4%	51.4%	78.9%

Table 3.6: WSD accuracies on SensEval/SemEval fine-grained and coarse-grained all-words tasks

Using the training data collected with the method described in Section 3.1.2, we apply our WSD system on the SensEval-2, SensEval-3, and SemEval-2007 English all-words tasks. Similarly, we also compare the performance of our system to the top three systems that participated in the above tasks (Palmer et al., 2001; Snyder and Palmer, 2004; Pradhan et al., 2007; Navigli, Litkowski, and Hargraves, 2007).

The evaluation results are shown in Table 3.6. We observe that IMS easily outperforms the WNs1 baseline. The differences of using different classifiers are not as significant as in the lexical-sample tasks. IMS with LIBLINEAR ranks first in SensEval-3 English fine-grained all-words task[9] and SemEval-2007 English coarse-

[9]The second best participating system in SensEval-3 English fine-grained all-words task is

grained all-words task. IMS using the WEKA toolkit ranks first in SemEval-2007 English fine-grained all-words task. No matter which classifier is applied, IMS is always competitive in the all-words tasks. It shows that the training data we collected is of high quality.

Overall, IMS achieves good WSD accuracies on both the SensEval/SemEval English lexical-sample tasks and all-words tasks. The performance of IMS shows that it is a state-of-the-art WSD system.

3.3 Summary

IMS is an English all-words WSD system. It provides a flexible platform for a supervised learning approach to WSD. It is also an all-words WSD component with good performance for other NLP applications.

The framework of IMS allows users to integrate different preprocessing tools to generate additional knowledge sources. Users can implement various feature types and different machine learning methods according to their requirements. By default, the IMS system implements three kinds of feature types and uses a linear kernel SVM as the classifier.

IMS achieves competitive performance in the evaluation on several SensEval/SemEval English lexical-sample tasks. We also evaluate IMS on the lexical sample tasks of three other languages: Italian, Spanish, and Chinese. Overall, IMS achieves state-of-the-art performance on all these languages, demonstrating its strength and robustness.

With this system, we also provide classification models pre-trained with the sense-annotated training examples from SemCor, the DSO corpus, and 6 parallel corpora, for a large number of the most frequent English content words. Evaluation on SensEval/SemEval English all-words tasks shows that IMS with these

SenseLearner, whose performance is significantly lower than IMS.

models achieves state-of-the-art WSD accuracies compared to the top participating systems.

Chapter 4

Domain Adaptation for Word Sense Disambiguation

In this chapter, we investigate domain adaptation for supervised learning systems for WSD. In our first experiment, we observe that supervised WSD systems trained with a large number of in-domain sense-annotated examples can obtain high level of accuracy, but they nevertheless suffer a substantial drop in accuracy when the training data is out-of-domain. This observation is consistent with previous works discussed in Section 2.2.3. Domain adaptation methods are needed to address this issue. We focus on the domain adaptation methods which use a few in-domain training examples. In particular, we apply the feature augmentation technique to WSD, which achieves good performance when a small amount of in-domain training data are available. To reduce the human annotation effort needed for acquiring the in-domain training data, we combine active learning and the feature augmentation technique to select in-domain examples to annotate and obtain half of the maximum increase in accuracy, by requiring only about 5% of the annotation effort.

In the next section, we first introduce the data set used in our experiments. Then we highlight the importance of domain adaptation for WSD as it substan-

tially affects the performance of a state-of-the-art WSD system when domain shifts in our first experiment. In Section 4.3 , we apply the feature augmentation technique to address the domain adaptation problem in WSD. We combine the feature augmentation technique in the active learning process to reduce the human effort needed for adaptation in Section 4.4. Finally, we conclude in Section 4.5.

4.1 Experimental Setting

In our experiments in this chapter, we use two data sets, OntoNotes as the in-domain data and SemCor as the out-of-domain data. As introduced in Section 1.2, SemCor is a one of the most widely used sense-annotated corpora. It is a portion of the Brown Corpus, which is a mixture of several genres such as scientific texts, fictions, etc. The first release of OntoNotes contains the sense annotations from the Wall Street Journal (WSJ) portion of the Penn Treebank (Marcus, Santorini, and Marcinkiewicz, 1993). Therefore, OntoNotes contains mainly business related news. In its first release (LDC2007T21) through the Linguistic Data Consortium (LDC), the project manually sense-annotated more than 40,000 examples belonging to hundreds of noun and verb types with an ITA of 90%, based on a coarse-grained sense inventory, where each word type has an average of only 3.2 senses.

The Wall Street Journal (WSJ) portion of the Penn Treebank corpus, where the annotated data of OntoNotes is drawn from, has been widely used in various NLP tasks such as syntactic parsing (Collins, 1999) and semantic role labeling (Carreras and Màrquez, 2005). These WSJ documents are divided into sections 00-24. In the previous studies, the practice is to use documents from WSJ sections 02-21 as training data, WSJ section 23 as test data, and the rest as development data. Table 4.1 illustrates the amount of sense-annotated data available from OntoNotes, across the various WSJ sections.[1] In the table, for each WSJ section, we list the

[1]We removed erroneous examples which were simply annotated with 'XXX' as sense-tag, or

Section	No. of word types	No. of word tokens	
		Individual	Cumulative
02	248	425	425
03	79	107	532
04	186	389	921
05	287	625	1546
06	224	446	1992
07	270	549	2541
08	177	301	2842
09	308	677	3519
10	648	3048	6567
11	724	4071	10638
12	740	4296	14934
13	749	4577	19511
14	710	3900	23411
15	748	4768	28179
16	306	576	28755
17	219	398	29153
18	266	566	29719
19	219	389	30108
20	288	536	30644
21	262	470	31114
23	685	3755	-

Table 4.1: Size of the sense-annotated data in the various WSJ sections.

number of word types, the number of sense-annotated examples, and the cumulative count on the number of sense-annotated examples. We follow previous work, using sense-annotated examples in sections 02-21 as training data and examples in section 23 as test data. Therefore, the in-domain training data in our experiments contains a total of slightly over 31,000 sense-annotated examples and the test data has 685 word types with more than 3,700 examples. Using the sense-annotated examples provided through OntoNotes, we conduct a large-scale WSD evaluation involving

annotated with senses that were not found in the sense-inventory provided. Also, since we will be comparing against training on SemCor later (which was annotated using WordNet senses), we removed examples annotated with OntoNotes senses which were not mapped to WordNet senses. On the whole, about 7% of the original OntoNotes examples were removed as a result.

hundreds of word types and tens of thousands of sense-annotated examples.

We use IMS as our WSD system, and train an individual classifier for each word type using the knowledge sources of local collocations, parts-of-speech (POS), and surrounding words. SVM with linear kernel implemented in WEKA is used as our learning algorithm, which was shown to achieve good WSD performance in (Lee and Ng, 2002; Chan, Ng, and Chiang, 2007).

4.2 In-Domain and Out-of-Domain Evaluation

In this section, we conduct two evaluations on OntoNotes data set. We first follow the common setting on WSJ corpus and perform an in-domain evaluation using data from OntoNotes. Next, we investigate the WSD performance when we train our system on sense-annotated examples gathered from a different domain, SemCor, as compared to the OntoNotes evaluation data.

4.2.1 Training and Evaluating on OntoNotes

Using examples from sections 02-21 as training data, we trained the classifier and evaluated on the examples from section 23. Both the training data and the test data are from the domain of WSJ corpus, so it is an in-domain test.

In our experiments, if a word type in section 23 has no training examples from sections 02-21, we randomly assign an OntoNotes sense to such a word occurrence. Under this experimental setting, our WSD system achieved an accuracy of 89.1%.

The 89.1% WSD accuracy we obtained is comparable to state-of-the-art syntactic parsing accuracies, such as the 91.0% performance by the statistical parser of (Charniak and Johnson, 2005). The high level of performance by syntactic parsers allows it to be used as an enabling technology in various NLP tasks. Thus, the fact that a state-of-the-art WSD system is able to achieve a high level of performance

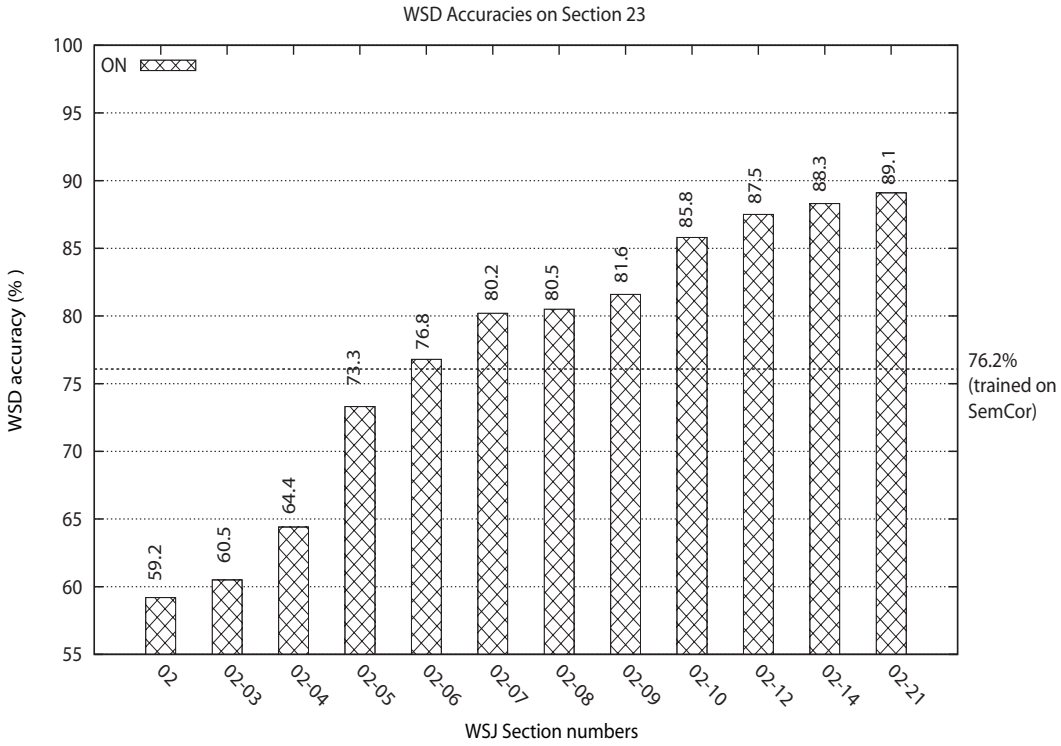

Figure 4.1: WSD accuracies evaluated on section 23, with different sections as training data.

means that such a WSD system will potentially be more usable for inclusion in NLP applications.

The high accuracy was achieved by training on a large amount (about 31,000) of manually sense annotated examples from sections 02-21 of the OntoNotes data. Besides training on the entire set of examples from sections 02-21, we investigate the performance achievable from training on various sub-sections of the data, as shown in Figure 4.1. The performance is improved from 59.2% with only section 02 as the training data to 89.1% with the entire training set. WSD accuracy increases as more training examples are added. This proves the importance of having a large amount of training data for WSD. However, the improvement on the right portion of Figure 4.1, where the size of the available training data is large, is not as huge as the left portion of the figure, where the size of the training data is small.

4.2.2 Using Out-of-Domain Training Data

Although our WSD system achieves a high accuracy of 89.1%, all the training data and test data are gathered from the same domain of WSJ. In reality, however, since manual sense annotation is time consuming, it is not feasible to collect such a large amount of manually sense-annotated data for every domain of interest. Hence, we need to investigate the performance of our WSD system when it is trained on out-of-domain data.

We employ SemCor as the out-of-domain training data and evaluate on section 23 of the OntoNotes corpus. As pointed out earlier, the training data set SemCor and test data set OntoNotes are from different domains.

For those word types in section 23 which do not have training examples from SemCor, we randomly chose an OntoNotes sense as the answer. Evaluating on the section 23 test data, our WSD system with SemCor as training data achieved only 76.2% accuracy. Compared to the 89.1% accuracy achievable when we trained on examples from sections 02-21, this is a substantially lower accuracy and a disappointing drop of performance and motivates the need for domain adaptation.

4.3 Concatenating In-Domain and Out-of-Domain Data for Training

The experiments in the last section show that a system trained with out-of-domain data is significantly worse than one trained with in-domain data. Thus when the training data and test data come from different domains, there is a necessity for domain adaptation. In this section, we perform domain adaptation experiments for WSD, focusing on domain adaptation methods that use some in-domain sense-annotated data. We first introduce the feature augmentation technique of (Daumé III, 2007), and then evaluate our WSD system with and without this technique.

4.3.1 The Feature Augmentation Technique for Domain Adaptation

The feature augmentation technique introduced by (Daumé III, 2007) is a simple yet very effective approach to domain adaptation. This technique is applicable when one has access to training data from the source domain and a small amount of training data from a new target domain.

Suppose we have data from n different domains $\{D_1, D_2, ..., D_n\}$. Assume $\boldsymbol{x}$ is an instance and its original feature vector is $\Phi(\boldsymbol{x})$. This technique essentially augments the feature space of the instance n times. The augmented feature vector for instance $\boldsymbol{x}$ is

$$\Phi'(\boldsymbol{x}) = \begin{cases} < \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}), \overbrace{\mathbf{0}, \mathbf{0}, ..., \mathbf{0}}^{n-1} > & \text{if } \boldsymbol{x} \in D_1 \\ < \Phi(\boldsymbol{x}), \mathbf{0}, \Phi(\boldsymbol{x}), \overbrace{\mathbf{0}, ..., \mathbf{0}}^{n-2} > & \text{if } \boldsymbol{x} \in D_2 \\ ..... & \\ < \Phi(\boldsymbol{x}), \overbrace{\mathbf{0}, ..., \mathbf{0}}^{i-1}, \Phi(\boldsymbol{x}), \overbrace{\mathbf{0}, ..., \mathbf{0}}^{n-i} > & \text{if } \boldsymbol{x} \in D_i \\ ..... & \\ < \Phi(\boldsymbol{x}), \overbrace{\mathbf{0}, ..., \mathbf{0}}^{n-1}, \Phi(\boldsymbol{x}) > & \text{if } \boldsymbol{x} \in D_n \end{cases},$$

where $\mathbf{0}$ is a zero vector of size $|\Phi(\boldsymbol{x})|$.

We see that the technique essentially treats the first field of the augmented feature space as holding general features that are not meant to be differentiated between different domains. Then, the other fields of the augmented feature space are reserved for holding the domain specific features.

During training and testing, the augmented features are used instead of the original features. The instances from the same domain share the same domain-specific features. Therefore, the augmented feature space can help distinguish different domains.

Despite its relative simplicity, this feature augmentation technique has been shown to outperform other domain adaptation techniques on various tasks such as named entity recognition, part-of-speech tagging, etc. (Daumé III, 2007)

In our experiment, we just have two domains, source domain D_s (out-of-domain) and target domain D_t (in-domain). Therefore, the feature vector $\Phi(\boldsymbol{x})$ of a WSD instance x will be augmented to:

$$\Phi'(\boldsymbol{x}) = \begin{cases} < \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}), \mathbf{0} > & \text{if } \boldsymbol{x} \in D_s \\ < \Phi(\boldsymbol{x}), \mathbf{0}, \Phi(\boldsymbol{x}) > & \text{if } \boldsymbol{x} \in D_t \end{cases}.$$

4.3.2 Experiments

As mentioned above, training our WSD system on SemCor examples gave a relatively low accuracy of 76.2%, as compared to the 89.1% accuracy obtained from training on OntoNotes section 02-21. Assuming we have access to some in-domain training data, then a simple method to potentially obtain better accuracies is to train on both the out-of-domain and in-domain examples. To investigate this, we concatenated the SemCor examples with different amounts of OntoNotes examples to train our WSD system. The obtained accuracies are shown as "SC+ON" in Figure 4.2. We also performed another set of experiments, where instead of simply concatenating the SemCor and OntoNotes examples, we applied the feature augmentation technique when concatenating these examples, treating SemCor examples as out-of-domain (source domain) data and OntoNotes examples as in-domain (target domain) data. We similarly show the resulting accuracies as "SC+ON Augment" in Figure 4.2.

Comparing the "SC+ON" and "SC+ON Augment" accuracies in Figure 4.2, we see that the feature augmentation technique *always* helps to improve the accuracy of our WSD system. Further, notice from the first few sets of results in the figure that when we have access to limited in-domain training examples from

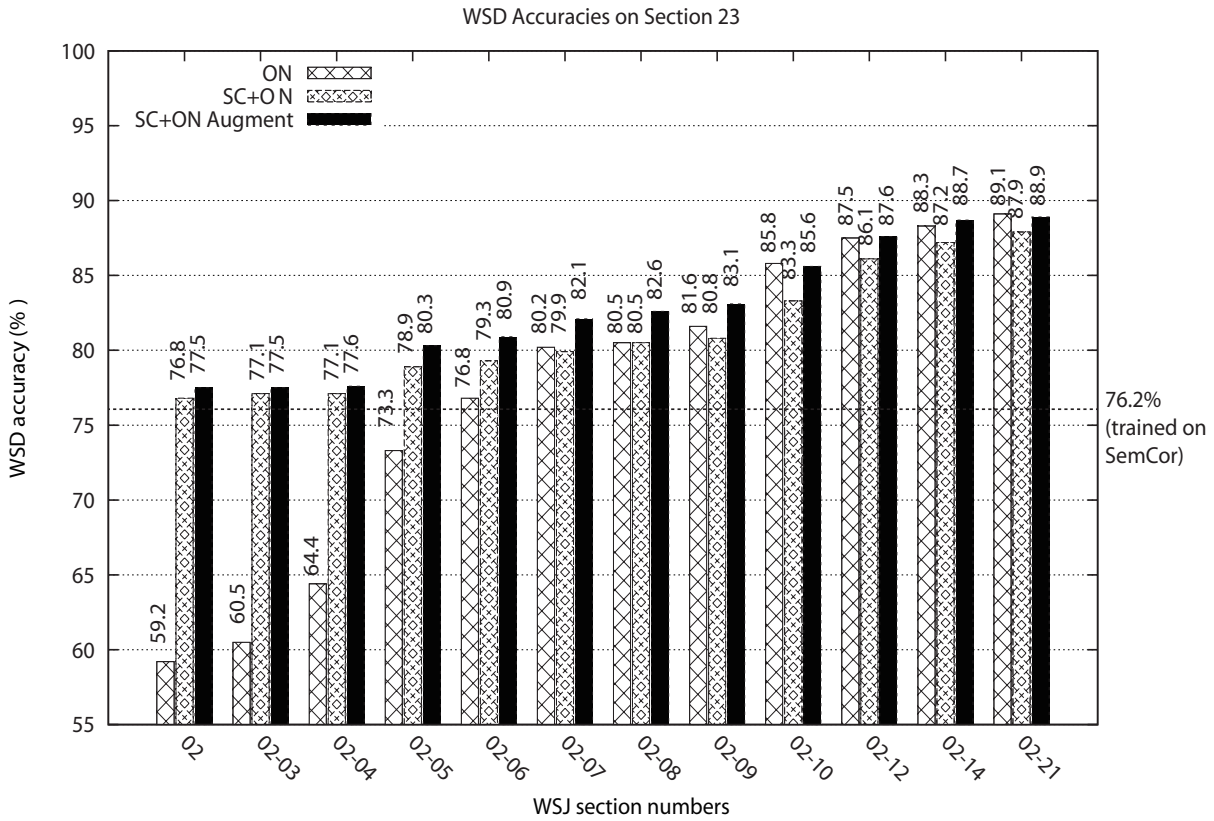

Figure 4.2: WSD accuracies evaluated on section 23, using SemCor and different OntoNotes sections as training data. ON: only OntoNotes as training data. SC+ON: SemCor and OntoNotes as training data, SC+ON Augment: Concatenating SemCor and OntoNotes via the Augment domain adaptation technique.

OntoNotes, incorporating additional training data from out-of-domain SemCor (either using the strategies "SC+ON" or "SC+ON Augment") achieves better accuracies than "ON".

Significance tests using one-tailed paired t-test reveal that these accuracy improvements are statistically significant at the level of significance 0.01 (all significance tests in the rest of this section use the same level of significance 0.01). This trend continues till the result for sections 02-06. These results validate the contribution of the SemCor examples.

The right half of Figure 4.2 shows the accuracy trend of the various strategies, in the unlikely event that we have access to a large amount of in-domain training examples. Although we observe that in this scenario, "ON" performs better than "SC+ON", it is still the case that "SC+ON Augment" continues to perform better than "ON" (where the improvement is statistically significant) till the result for sections 02-09. Beyond that, as we add more OntoNotes examples, significance testing reveals that the "SC+ON Augment" and "ON" strategies give comparable performance. This shows that the "SC+ON Augment" strategy, besides giving good performance when one has few in-domain examples, does continue to perform well even when one has a large number of in-domain examples.

4.4 Active Learning for Domain Adaptation

The experiment results in the last section show that when we have access to some in-domain examples, a good strategy is to concatenate the out-of-domain and in-domain examples via the feature augmentation technique. This suggests that when one wishes to apply a WSD system to a new domain of interest, it is worth the effort to annotate a small number of examples gathered from the new domain. However, instead of randomly selecting in-domain examples to annotate, we could use active learning (Lewis and Gale, 1994) to help select in-domain examples to annotate. By

doing so, we minimize the manual annotation effort needed for domain adaptation.

In WSD, several prior research efforts have successfully used active learning to reduce the annotation effort required (Fujii et al., 1998; Chen et al., 2006; Chan and Ng, 2007; Zhu and Hovy, 2007). With the exception of (Chan and Ng, 2007) which tried to adapt a WSD system trained on the BC part of the DSO corpus to the WSJ part of the DSO corpus, the other researchers simply applied active learning to reduce the annotation effort required and did not deal with the issue of adapting a WSD system to a new domain. Also, these prior research efforts only experimented with a few word types.

In contrast, in this section we perform active learning experiments on the hundreds of word types in the OntoNotes data, with the aim of adapting our supervised learning system for WSD trained on SemCor to the WSJ domain represented by the OntoNotes data.

4.4.1 Active learning with the Feature Augmentation Technique for Domain Adaptation

In the active learning algorithm, a set of sense-annotated training examples and a pool of unannotated examples are used. In each iteration, a system is trained on the sense-annotated examples. Then the system is applied on the unannotated example pool to select one or several representative examples with some example selection strategy. The senses of these selected examples are annotated by humans. Together with the human annotations, these examples are finally added back to the training data set for the next iteration.

Different from the common active learning algorithm, our initial training examples E_s and the pool of unannotated adaptation examples E_a are from different domains. E_s is from the source domain and E_a is from the target domain. In addition, the example selected in each iteration is added into a set of in-domain

```
E_s ← the set of SemCor training examples
E_a ← the set of OntoNotes sections 02-21 examples
E_t ← empty
while E_a ≠ φ
   p_min ← ∞
   Γ ← WSD system trained on E_s and E_t using the feature augmentation technique
   for each d ∈ E_a do
      ŝ ← word sense prediction for d using Γ
      p ← confidence of prediction ŝ
      if p < p_min then
         p_min ← p, d_min ← d
      end
   end
   E_a ← E_a − {d_min}
   provide correct sense s for d_min and add d_min to E_t
end
```

Figure 4.3: The active learning algorithm.

sense-annotated examples E_t, instead of adding to E_s. As shown in Figure 4.3, we train an initial WSD system using only the set E_s. We then apply our WSD system on all the examples in set E_a. Using the *uncertainty sampling* strategy (Lewis and Gale, 1994) for example selection, the example in E_a which is predicted with the lowest confidence will be removed from E_a and annotated with the correct sense. Then it is added to E_t, the set of in-domain examples that have been selected via active learning thus far. Since we have found that the feature augmentation technique is useful in increasing WSD accuracy instead of simply mixing the examples in E_s and E_t, we will apply the feature augmentation technique to concatenate the source domain training examples in E_s and the selected adaptation examples in E_t to train a new WSD system, which is then applied again on the set E_a of remaining adaptation examples. This active learning process continues until we have used up all the adaptation examples.

4.4.2 Experiments

For our experiments, the SemCor examples form our initial set of training examples E_s, while the OntoNotes examples from sections 02-21 will be used as our pool of adaptation examples E_a, from which we will select examples to annotate via active learning. Note that because we are using OntoNotes sections 02-21 (which have already been sense-annotated beforehand) as our adaptation data, the annotation of the selected example during each active learning iteration is simply simulated by referring to its annotated sense.

We perform active learning experiments on *all* the word types that have sense-annotated examples from OntoNotes sections 02-21, and show the evaluation results on OntoNotes section 23 as the topmost "all" curve in Figure 4.4. Since our aim is to reduce the human annotation effort required in adapting a WSD system to a new domain, we may not want to perform active learning on all the word types in practice. Instead, we can maximize the benefits by performing active learning only on the more frequently occurring word types. Hence, in Figure 4.4, we also show via various curves the results of applying active learning only to various sets of word types, according to their frequency, or number of sense-annotated examples in OntoNotes sections 02-21. Note that the various accuracy curves in Figure 4.4 are plotted in terms of evaluation accuracies over all the test examples in OntoNotes section 23, hence they are directly comparable to the results reported thus far in this section. Also, since the accuracies for the various curves stabilize after 35 active learning iterations, we only show the results of the first 35 iterations.

From Figure 4.4, we note that by performing active learning on the set of 150 most frequently occurring word types, we are able to achieve a WSD accuracy of 82.6% after 10 active learning iterations. Comparing to using only the out-of-domain SemCor examples, we have gained a 6.4% absolute improvement in accuracy (82.6% − 76.2%) by just using 1,500 in-domain OntoNotes examples.

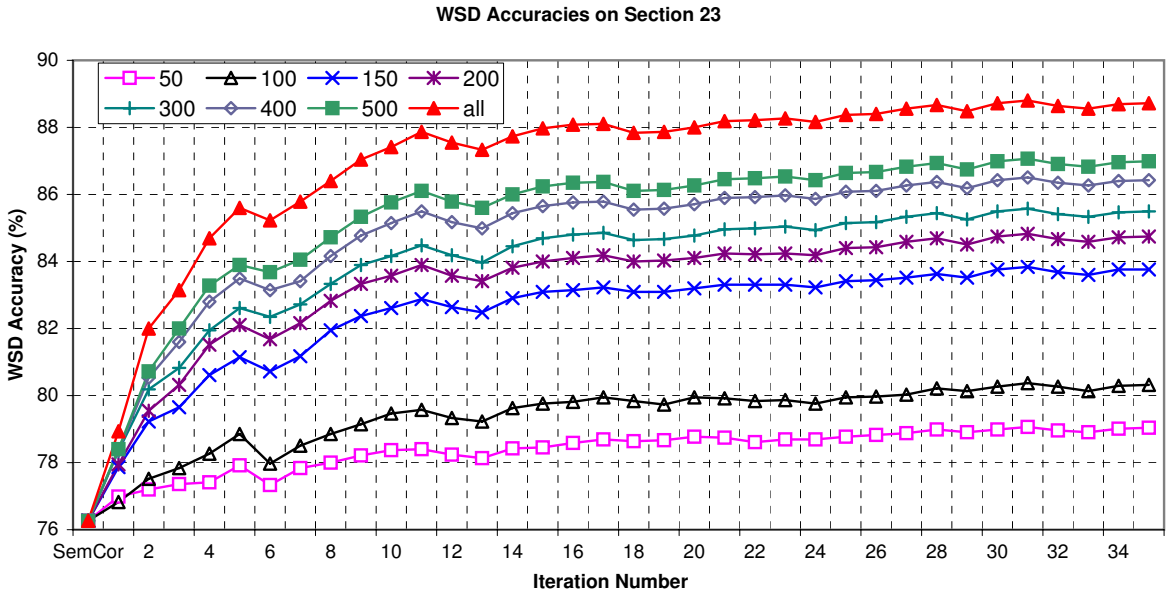

Figure 4.4: Results of applying active learning with the feature augmentation technique on different number of word types. Each curve represents the adaptation process of applying active learning on a certain number of most frequently occurring word types.

Compared with the 12.9% (89.1% − 76.2%) improvement in accuracy achieved by using all 31,114 OntoNotes sections 02-21 examples, we have obtained half of this maximum increase in accuracy, by requiring only about 5% (1,500/31,114) of the total number of sense-annotated examples. In another experiment, we randomly select 10 OntoNotes examples for the set of 150 word types. With these 1,500 randomly selected examples, we achieve an accuracy of 81.7%, which is significanly lower than 82.6% achieved with active learning. Based on these results, we propose that when there is a need to apply a previously trained WSD system to a different domain, one can apply the feature augmentation technique with active learning on the most frequent word types, to greatly reduce the annotation effort required while obtaining a substantial improvement in accuracy.

4.5 Summary

In this section, we investigated the domain adaption problem for WSD. Our experiments on OntoNotes show that the shift of domain causes a drop in performance of a supervised learning system for WSD by more than 10% in accuracy. We applied the feature augmentation technique and active learning to perform domain adaptation of our WSD system and obtained half of the increase in accuracy by only annotating 5% of the in-domain examples. Our experiments show that the feature augmentation technique combined with active learning can greatly reduce the human annotation effort needed for domain adaptation and achieve a substantial improvement when adapting a WSD system to a new domain.

Chapter 5

Automatic Extraction of Training Data from Parallel Corpora

As introduced in Section 2.2.2, different senses of an English word often have distinct translations in a second language. Thus it is possible to identify the sense of a word in context if its translation is known. Because the non-Indo-European languages, such as Basque, Chinese, and Turkish, have a higher probability to differently lexicalize English senses (Resnik and Yarowsky, 2000), Chan and Ng (2005a) proposed a method to extract training examples from English-Chinese parallel corpora with manually annotated Chinese translations for each English word sense.

Compared to sense-annotating training examples directly, the human effort needed in the approach of Chan and Ng (2005a) is relatively reduced. However, in WSD, different sense-annotated data are needed for different word types. Considering the huge number of word types in a language, manually assigning translations to the senses of words still needs a large amount of human effort. If we can find a completely automatic way to collect such translations in a second language for senses of a word, the whole process of extracting training examples from parallel texts for WSD will be completely unsupervised.

In this chapter, we adopt the approach of (Chan and Ng, 2005a) to extract training examples from parallel corpora and extend their work by proposing four methods to find Chinese translations for English WordNet senses without any additional human effort.

The organization of this chapter is as follows. In Section 5.1, we introduce the method of extracting training data from parallel corpora proposed by (Chan and Ng, 2005a). In Section 5.2, we describe our methods of gathering Chinese translations automatically for English senses. We first extract the Chinese translations from an English-Chinese bilingual WordNet and an English-Chinese bilingual dictionary in subsection 5.2.1 and subsection 5.2.2, respectively. In the next two subsections, we propose two methods to gather more Chinese translations that do not appear in dictionaries but appear in bilingual corpora. With the above four methods, the selection of Chinese translations is done without any additional manual human effort. As such, the entire process of extracting training data for WSD from parallel corpora is *fully automatic* and *unsupervised*. The quality of the extracted Chinese translations and their impact on WSD are evaluated in Section 5.3, followed by a summary in Section 5.4.

5.1 Acquiring Training Data from Parallel Corpora

This section describes the method of extracting training data from parallel corpora. We assign valid Chinese translations to each sense of an ambiguous English word. Suppose a Chinese word c is a valid translation of sense s of an English word e. The process of extracting training examples for e from parallel texts is as follows:

- Collect sentence-aligned parallel texts.

- Perform tokenization on the English texts with the Penn Treebank tokenizer;
- Perform Chinese word segmentation on the Chinese texts with the Chinese word segmentation method proposed by Low *et al.* (2005);
- Perform word alignment on the parallel texts using the GIZA++ software (Och and Ney, 2000);
- Suppose an occurrence o of e is aligned to c and c is a valid translation of sense s of e. This occurrence o is then labeled with sense s.
- Extract occurrences of e and their 3-sentence surrounding contexts as sense-annotated training data.

#	Translations	Sense descriptions
1	文章	Nonfictional prose forming an independent part of a publication
2	物品 物件 货品	One of a class of artifacts
3	条文 条款 条	A separate section of a legal document
4	冠词	A determiner that may indicate the specificity of reference of a noun phrase

Table 5.1: Senses of the noun "article" in WordNet

For example, Table 5.1 shows the four senses of the noun "article" in WordNet. The first column is the sense number, the second column lists the valid Chinese translations, and the last column lists the sense descriptions. Given the following sentence pair in parallel texts:

- The reporter wrote an article about environment protection.
- 记者 写了 一篇 关于 环境 保护 的 文章。

according to the output of GIZA++, "article" is aligned to 文章, which has been selected as the Chinese translation for sense 1. Therefore, this instance of "article" will be tagged as sense 1.

In the experiment of (Chan and Ng, 2005a), they used six English-Chinese parallel corpora: Hong Kong Hansards, Hong Kong News, Hong Kong Laws, Sinorama, Xinhua News, and the English translation of Chinese Treebank. These corpora are all available from the Linguistic Data Consortium (LDC). With WordNet as the sense inventory, (Chan and Ng, 2005a) manually assigned Chinese translations to the top 60% most frequently occurring noun types in the Brown corpus.

Parallel corpora	Size of texts (million words (MB))	
	English texts	Chinese texts
Hong Kong Hansards	39.9 (223.2)	35.4 (146.8)
Hong Kong News	16.8 (96.4)	15.3 (67.6)
Hong Kong Laws	9.9 (53.7)	9.2 (37.5)
Sinorama	3.8 (20.5)	3.3 (13.5)
Xinhua News	2.1 (11.9)	2.1 (8.9)
English translation of Chinese Treebank	0.1 (0.7)	0.1 (0.4)
Total	72.6 (406.4)	65.4 (274.7)

Table 5.2: Size of English-Chinese parallel corpora

In our experiment, we follow the work of (Chan and Ng, 2005a) and also use the six parallel corpora listed above. Table 5.2 lists the statistics of the 6 English-Chinese parallel corpora. Different from the work of (Chan and Ng, 2005a) in which they used manually assigned Chinese translations, we present our methods of automatically collecting the Chinese translations in the next section.

5.2 Automatic Selection of Chinese Translations

In this section, we propose four methods to collect Chinese translations for the WordNet senses of English words. We first describe the method of using an English-Chinese bilingual WordNet. Then we explain the usage of common bilingual English-Chinese dictionaries. Because Chinese translations collected from dictionaries may not be of use in extracting training data from parallel corpora, we

also propose two additional methods: shortening the collected Chinese translations and finding their synonyms.

5.2.1 Academia Sinica Bilingual Ontological WordNet

We first extract Chinese translations from a Chinese version of WordNet, the Academia Sinica Bilingual Ontological WordNet (BOW) (Huang, Chang, and Lee, 2004). BOW is a bilingual dictionary which integrates the English WordNet and two other resources, Suggested Upper Merged Ontology (SUMO) and the English-Chinese Translation Equivalents Database (ECTED). In BOW, the English WordNet was manually mapped to SUMO and ECTED. With the integration of these three resources, BOW functions as an English-Chinese bilingual WordNet. That is, each WordNet synset has a set of corresponding Chinese translations in BOW.

After carrying out some preprocessing, we extract 94,874 Chinese translations from BOW for all of the 66,025 WordNet noun synsets. For example, in WordNet-1.6, synset "00601680.n" (approach, attack, plan of attack) means "a formulation adopted in tackling a problem; the Chinese translations extracted for this synset are "手段" and "方法".

5.2.2 A Common English-Chinese Bilingual Dictionary

BOW provides Chinese translations for all WordNet synsets, but each noun synset has only 1.4 Chinese translations on average. As reported in our evaluation results, these Chinese translations available in BOW are not adequate for us to extract sufficient training examples from parallel corpora. As such, we propose a method to extract more Chinese translations for WordNet synsets from common English-Chinese bilingual dictionaries.

In English-Chinese bilingual dictionaries, a set of Chinese translations are provided for each English word sense. However, the sense definitions and granular-

ities in these dictionaries can be quite different from WordNet. Thus, it is hard to map the English word senses in these common dictionaries to WordNet senses. We propose two heuristics to make use of the Chinese translations provided by common bilingual dictionaries:

1. If two or more English synonyms in a WordNet synset *syn* share the same Chinese translation c in a bilingual dictionary, we assign c as a Chinese translation for synset *syn*.

2. Suppose an English word e is monosemous in WordNet. Let *syn* be the WordNet synset corresponding to the only sense of e. Then all Chinese translations of e from a bilingual dictionary are assigned as the Chinese translations for synset *syn*.

In our experiment, we use an English-Chinese bilingual dictionary, Kingsoft PowerWord 2003.[1] PowerWord 2003 contains Chinese translations of English sense entries in the American Heritage Dictionary. For an English word sense, PowerWord lists a set of Chinese translations. Similar to other common dictionaries, the sense definitions of PowerWord and WordNet are different and the Chinese translations in PowerWord cannot be directly mapped to WordNet senses. The following two examples show how the above two heuristics make use of the Chinese translations from Powerword:

1. In WordNet 1.6, synset "10969750.n", which means "a time interval during which there is a temporary cessation of something", has 5 synonyms: *pause*, *intermission*, *break*, *interruption*, and *suspension*. In PowerWord, *pause* and *suspension* have the same Chinese translation "中止"; *break*, *pause*, and *suspension* share the same Chinese translation "暂停". According to the first

[1] http://www.iciba.com/

heuristic, “中止” and “暂停” are assigned as Chinese translations to synset “10969750.n”.

2. In WordNet 1.6, synset “10382904.n”, which means “a desirable state”, has two synonyms: blessing and boon. Because the noun *boon* is monosemous in WordNet, all Chinese translations of *boon* “恩惠”, “实惠”, and “福利” in PowerWord are assigned to synset “10382904.n”.

Via the above two ways, 52,599 Chinese translations are extracted from PowerWord for 29,066 out of 66,025 noun synsets. On average, each English synset has 1.8 Chinese translations.

So far, Chinese translations are gathered from both BOW and PowerWord for WordNet synsets. For each English word e, we can find the Chinese translations for its senses by referring to their corresponding synsets. Because WordNet senses are ordered such that a more frequent sense appears before a less frequent one, if several senses of e share an identical Chinese translation c, only the sense with the smallest sense number (corresponding to the most frequently occurring sense) among these senses will have c assigned as a translation. In this way, a Chinese translation c is only assigned to one sense of a word e.

5.2.3 Shortening Chinese Translations

The Chinese translations from dictionaries sometimes contain modifiers. Thus these translations are usually long and may have no occurrences in parallel texts aligned to the corresponding English words. In this case, no training examples can be extracted from parallel texts with such Chinese translations. For instance, the Chinese translation “尤指国家的税收” (especially referring to federal tax) extracted from dictionary for the second WordNet sense of *revenue* is not aligned to the English

word *revenue* in parallel texts. As a result, no training examples for *revenue* will be extracted with this Chinese translation. But as a good Chinese definition for sense 2 of *revenue*, "尤指国家的税收" is supposed to contain some useful information related to *revenue*. In fact, we can discard the modifiers of this translation and only keep the last two Chinese characters, "税收" (tax), which is a good translation for *revenue*.

In this subsection, we propose a method to make use of the Chinese translations those have no occurrences aligned to their corresponding English words in parallel texts, by shortening them. Suppose sense s of an English word e has such a Chinese translation c from dictionary. We first generate its longest prefix *pre* and longest suffix *suf* which happen to align to e in parallel texts. *pre* and *suf*, if found, are the possible shortened candidate translations of c that may be selected as translations of s. Among these shortened translation candidates, we further discard a candidate if it is a substring of any Chinese translations from dictionary for a different sense s' of e. The remaining translation candidates are selected for use. Each chosen prefix or suffix of c is a Chinese translation of the sense s associated with c.

Using this method, for the above example, we generate a shortened Chinese translation "税收" (tax) for "尤指国家的税收" for the second sense of *revenue* in WordNet. Similarly, the Chinese translation "价值观念" (value concept), for sense 6 of the English noun *value*, has no occurrences aligned to *value* in parallel texts. By applying this method, we get two shortened Chinese translations "价值观" (value concept) and "观念" (concept).

5.2.4 Using Word Similarity Measure

With the methods proposed in the previous sections, we collect Chinese translations from the dictionaries BOW and PowerWord, and their prefixes and suffixes.

Define *selected*(e) as the set of Chinese translations selected for an English word e (associated with any of its senses). The occurrences of a Chinese translation c in parallel texts which are aligned to e will be extracted as training examples for e if and only if $c \in$ *selected*(e). Accordingly, if a Chinese translation c does not belong to *selected*(e), its occurrences in parallel texts that are aligned to e will be wasted.

In this subsection, we propose a method to assign Chinese translations which are not in *selected*(e), but have occurrences aligned to e in parallel texts, to appropriate senses by measuring their similarities with Chinese translations in *selected*(e). The assumption of this method is that two Chinese words are synonymous if they have the same translation and their distributional similarity is high.

5.2.4.1 Calculating Chinese Word Similarity

We use the distributional similarity measure based on syntactic relations as described in Lin (1998) as our word similarity measure. Suppose (w, r, m) is a dependency triple extracted from a corpus parsed by a dependency parser, where r is the dependency relation, w is the head word, and m is the modifier together with its part-of-speech. Define $||w, r, m||$ as the frequency count of the dependency triple (w, r, m) in a parsed corpus. If w, r, or m is a wild card '*', the frequency count will be the sum of frequency counts of all the dependency triples that match the rest of the expression. Define $I(w, r, m)$ as the amount of information contained in (w, r, m), whose value is

$$I(w, r, m) = \log \frac{||w, r, m|| \times ||*, r, *||}{||w, r, *|| \times ||*, r, m||}.$$

Let $T(w)$ be the set of pairs (r, m) such that $I(w, r, m)$ is positive. The similarity $sim(w_1, w_2)$ between two words w_1 and w_2 is calculated as

$$sim(w_1, w_2) = \frac{\sum_{(r,m) \in T(w_1) \cap T(w_2)} (I(w_1, r, m) + I(w_2, r, m))}{\sum_{(r,m) \in T(w_1)} I(w_1, r, m) + \sum_{(r,m) \in T(w_2)} I(w_2, r, m)} \tag{5.1}$$

We first train the Stanford parser (de Marneffe, MacCartney, and Manning, 2006) on Chinese Treebank 5.1 (LDC2005T01U01), and then parse the Chinese side of the 6 parallel corpora with the trained parser to output dependency parses.[2] The whole parsing process takes about 300 CPU hours on a 2.83GHz CPU. We only consider the triples of subject relation, direct object relation, and modifying relation. Dependency triples whose head word's frequency is less than 10 are removed. From the parsed corpus, we extract a total of 13.5 million dependency triples. The similarity between two Chinese words is calculated using the above similarity measure on the set of 13.5 million dependency triples.

5.2.4.2 Assigning Chinese Translations to English Senses Based on Word Similarity

Suppose e is an English word, and c is a Chinese translation of e. Define $sense(c)$ as the sense of e that c is assigned to, and $count(c)$ as the number of occurrences of c aligned to e in the parallel corpora. The function avg calculates the average value of a set of values, and the function σ calculates the standard deviation of a set of values.

Figure 5.1 shows the process in which we assign the set of Chinese translations Φ that are aligned to e in parallel texts but not selected as Chinese translation for e in our previous methods. Because most of the Chinese translations aligned to e with low frequency are erroneous in the word alignment output of GIZA++, in the first step, we eliminate the Chinese translations in Φ whose occurrence counts are below the average. For each Chinese translation c remaining in Φ, we calculate its similarity scores with the Chinese translations in $selected(e)$. Suppose c_{max} is the Chinese translation in $selected(e)$ which c is most similar to. We consider c as a candidate Chinese translation for the sense associated with c_{max}. To ensure

[2]Due to computational consideration, all sentences that are longer than 50 words are not included.

$\Phi \leftarrow$ the set of Chinese translations that are aligned to e in parallel texts but not in $selected(e)$
$count_{avg} \leftarrow avg(\{count(c) : c \in \Phi\})$
for each $c \in \Phi$
  if $count(c) < count_{avg}$
    $\Phi \leftarrow \Phi - \{c\}$
    continue
  end if
  $S[c] \leftarrow \max_{c' \in selected(e)} sim(c, c')$
  $C[c] \leftarrow argmax_{c' \in selected(e)} sim(c, c')$
end for
$threshold \leftarrow \min(avg(S) + \sigma(S), \theta)$
for each $c \in \Phi$
  if $S[c] \geq threshold$
    set c as a Chinese translation for $sense(C[c])$
  end if
end for

Figure 5.1: Assigning Chinese translations to English senses using word similarity measure.

that c is a Chinese synonym of c_{max}, we require that the similarity score between c and c_{max} should be high enough. A threshold $arg(S) + \sigma(S)$ is set to filter those candidates with low scores, where $arg(S) + \sigma(S)$ is the mean plus standard deviation of the scores of all candidates. To ensure that $arg(S) + \sigma(S)$ is not too high such that most of the candidates are filtered out, we set an upper bound θ for the threshold. In our experiment, θ is set to be 0.1. Finally, each candidate whose score is higher than or equal to the threshold will be assigned to the sense of its most similar Chinese translation.

For example, using our method, "方式", "手法", "办法", "形式", and "模式" are correctly assigned to the first sense of *approach* (ideas or actions intended to deal with a problem or situation), because they are similar to the Chinese translations "手段" or "方法" extracted from dictionary according to the similarity measure. Similarly, sense 3 of *judgement* (the determination by a court) has the Chinese

translation “判决” from dictionary, thus “裁决” which is a synonym of “判决” is assigned to this sense.

5.3 Evaluation

In this section, we first manually check the quality of the Chinese translations gathered with the above methods. We then evaluate the training examples extracted from parallel texts with these Chinese translations on OntoNotes data set.

5.3.1 Quality of the Automatically Selected Chinese Translations

In Section 5.2.2, Chinese translations are extracted from PowerWord for WordNet synsets in two ways. We manually evaluate 100 randomly selected synsets which get extended Chinese translations with the first way. 134 out of 158 (84.8%) extended Chinese translations in these 100 synsets are found to be good translations. Similarly, 100 synsets, which get extended Chinese translations from PowerWord with the second way, are randomly selected for evaluation. 214 out of 261 (82.0%) extended Chinese translations in these synsets are good. Chinese translations from dictionaries are shortened with the method described in Section 5.2.3. We manually evaluate 50 randomly selected Chinese translations, and find that 70% (35/50) of these shortened Chinese translations are appropriate. In Section 5.2.4, we extend the Chinese translations of each English word by finding Chinese synonyms. 329 Chinese synonyms of 100 randomly selected English words which get Chinese translations in this method are manually evaluated. About 77.8% (256/329) of them are found to be good Chinese translations.

We also manually evaluate 500 randomly selected sense-tagged instances from parallel texts for 50 word types (10 instances for each word type). The accu-

racy of these sample instances is 80.4% (402/500).

5.3.2 Experiments on OntoNotes

To measure the effect of these training examples on WSD, we evaluated some combinations of the above translation selection methods on all noun types in OntoNotes 2.0 data. We used IMS as our WSD system in the experiment, with POS tags, local collocations, and surrounding words as features and SVM with linear kernel in Weka as classifier. The test data set OntoNotes 2.0 (LDC2008T04) contains nearly 83,500 sense-annotated examples belonging to hundreds of noun and verb types. Table 5.3 lists some statistics of nouns in OntoNotes 2.0. There are 605 noun types with 29,510 noun tokens in OntoNotes 2.0. These nouns have 3.5 senses on average. Among the top 60% most frequent nouns with manually annotated Chinese translations from (Chan and Ng, 2005a), 257 of them have sense-annotated examples in our test data set. We refer to this set of 257 nouns as *T60Set*. The nouns in this set have a higher average number of senses (4.3).

Noun Set	No. of noun types	Average no. of senses	No. of noun tokens
T60Set	257	4.3	22,353
All nouns	605	3.5	29,510

Table 5.3: Statistics of sense-annotated nouns in OntoNotes 2.0

In the experiment, training examples with WordNet senses are mapped to OntoNotes senses. One of our baselines is the strategy "WNs1". Because the first sense in WordNet is the most frequent sense of a word on the SemCor corpus, in "WNs1", we always assign the OntoNotes sense that is mapped to the first sense in WordNet as the answer to each noun token. As mentioned previously, SemCor is the most widely used sense-annotated corpus. We use the strategy "SC", which uses only the SemCor examples as training data, as another baseline of supervised

systems.

Using the Chinese translations collected with our proposed methods, in the following strategies, a maximum of 1,000 examples gathered from parallel texts are merged with the SemCor examples for each noun type:

- strategy "SC+BOW" uses Chinese translations from BOW to extract examples from parallel texts for all noun types;
- strategy "SC+Dict" uses the Chinese translations from both BOW and PowerWord;
- strategy "SC+Dict+Sht" applies the method described in Section 5.2.3 to extend the Chinese translations in strategy "SC+Dict";
- strategy "SC+Dict+Sht+Sim" extends the Chinese translations in strategy "SC+Dict+Sht" using the method described in Section 5.2.4;
- strategy "SC+Manu" only extracts training examples from parallel texts for the noun types in *T60Set* with their manually annotated Chinese translations.

Strategy	Evaluation Set	
	T60Set	All nouns
SC+Manu	80.3%	77.0%
SC+Dict+Sht+Sim	77.7%	75.4%
SC+Dict+Sht	77.1%	74.9%
SC+Dict	76.7%	74.3%
SC+BOW	76.2%	73.7%
SC	73.9%	72.2%
WNs1	76.2%	73.5%

Table 5.4: WSD accuracy on OntoNotes 2.0

For each noun type, the examples from the parallel corpora are randomly chosen according to the sense distribution of that noun in SemCor corpus. When

we use the Chinese translations automatically selected to gather training examples from parallel texts, we prefer the examples related to the Chinese translations from dictionary BOW and PowerWord. If a word type has no training data, a random OntoNotes sense will be selected as the answer.

Table 5.4 shows the WSD accuracies of different strategies on *T60Set* and all nouns in OntoNotes 2.0. Comparing to WNs1 baseline, all the strategies using training examples from parallel texts achieve higher or comparable accuracies on both *T60Set* and all nouns.

Strategy	Evaluation Set	
	T60Set	All nouns
SC+Manu	24.5%	17.3%
SC+Dict+Sht+Sim	14.6%	11.5%
SC+Dict+Sht	12.3%	9.7%
SC+Dict	10.7%	7.6%
SC+BOW	8.8%	5.4%

Table 5.5: Error reduction comparing to *SC* baseline

In Table 5.5, we list the error reduction rate of the supervised learning strategies compared to the supervised baseline strategy "SC". Comparing to the supervised baseline "SC", our approach "SC+Dict+Sht+Sim" achieves a 3.8% absolute improvement in accuracy for *T60Set* and a 3.2% absolute improvement in accuracy for *All nouns*. That is, our *completely automatic* approach is able to obtain more than half (59%) of the improvement obtained using the manual translation assignment approach of "SC+Manu" for *T60Set*, and 67% of the improvement for *All nouns*.

Moreover, to check whether one strategy is statistically significantly better than another, we conducted one-tailed paired t-test with a significance level $p = 0.01$. The t statistic of the difference between each test example pair is computed. The significance test results on all noun types in OntoNotes 2.0 are given in Figure

SC+Manu > SC+Dict+Sht+Sim
> SC+Dict+Sht
> SC+Dict
> SC+BOW ∼ WNs1
> SC

Figure 5.2: Significance test results on all noun types.

5.2.

Because the significance tests on the *T60Set* have similar results, we will discuss the significance test results without differentiating these two sets of noun types. The "WNs1" baseline is only significantly better than strategy "SC". It is comparable to strategy "SC+BOW" but significantly worse than the other strategies. In each step where we extend the automatic Chinese translation selection, a significant improvement is achieved in the WSD accuracy. The results confirm the high quality of the automatically selected Chinese translations. Although strategy "SC+Manu" is still significantly better than all other strategies, the improvement achieved by using automatically selected Chinese translations is considerable, given that no human effort is needed.

5.4 Summary

The bottleneck of current supervised WSD systems is the lack of sense-annotated training data. In this chapter, we extend the method of Chan and Ng (2005a) by automatically selecting Chinese translations for English senses. With our approach, the process of extracting sense-annotated examples from parallel texts does not need any extra human effort. Evaluation on a large number of noun types in OntoNotes 2.0 data shows that the training examples gathered with our approach are of high quality, and results in statistically significant improvement in WSD accuracy.

Chapter 6

Word Sense Disambiguation for Information Retrieval

As discussed in Section 2.5.2, in the application of WSD to IR, researchers arrived at conflicting observations and conclusions. Positive reports show that WSD can bring two kinds of benefits to IR systems:

- First, queries may contain ambiguous words (terms), which have multiple meanings[1]. The ambiguities of these query words can hurt retrieval precision. Identifying the correct meaning of the ambiguous words in both queries and documents can help improve retrieval precision.

- Second, query words may have tightly related meanings with other words not in the query. Making use of these relations between words can improve retrieval recall.

Overall, IR systems can potentially benefit from the correct meanings of words provided by WSD systems. However, the WSD effect is not significant because

[1]In our analysis of the ambiguity of query terms, we assume each query has only one interpretation. Sanderson (2008) demonstrated that ambiguous queries, which have multiple interpretations, also impact IR effectiveness. However, our current work focuses on dealing with queries that have only one interpretation.

query terms usually have a skewed sense distribution and the collocation effect of other query terms already performs some disambiguation. Moreover, errors made by an automated WSD component can easily neutralize its positive effect and hurt IR performance. Thus it is important to reduce the negative impact of erroneous disambiguation, and the integration of senses into a traditional term index, such as a stem-based index, is one possible solution. The utilization of semantic relations has proved to be helpful for IR, thus it is also interesting to investigate the utilization of semantic relations between senses.

In this chapter, we investigate the use of word senses to improve the performance of IR. We build a WSD system based on supervised learning from parallel corpora without manual sense annotation, and propose an approach to automatically assign senses to words in short queries. With senses assigned to both queries and documents, we then incorporate the senses into the language modeling (LM) approach to IR (Ponte and Croft, 1998) by adjusting term frequency. We utilize sense synonym relations to further improve the performance of our IR system. Our evaluation on standard *TREC*[2] data sets shows that supervised WSD outperforms two other WSD baselines and can significantly improve IR accuracy.

The remainder of this chapter is organized as follows. Section 6.1 introduces the LM approach to IR, a pseudo relevance feedback method, and a collection enrichment technique. We describe our WSD system which is built based on parallel corpora with translations as senses, and our approach of generating word senses for query terms in Section 6.2. In Section 6.3, we introduce our novel method of incorporating word senses and their synonyms into the LM approach. Next, we present experimental results on four TREC query sets and analyze the results in Section 6.4. Finally, a summary is given in Section 6.5.

[2]http://trec.nist.gov/

6.1 The Language Modeling Approach to IR

In this section, we first describe the LM approach to IR. Then we introduce a pseudo relevance feedback method as well as a collection enrichment technique.

6.1.1 The Language Modeling Approach

In the language modeling approach to IR, language models are constructed for each query q and each document d in a text collection C. The documents in C are ranked by the distance to a given query q according to the language models. The most commonly used language model in IR is the unigram model, in which terms are assumed to be independent of each other. In the rest of this chapter, language model will refer to the unigram language model.

One of the commonly used measures of the similarity between query model and document model is negative Kullback-Leibler (KL) divergence (Lafferty and Zhai, 2001):

$$Rank(d|q) \propto -D(\theta_q||\theta_d)$$

The higher the similarity, the higher the rank. With the unigram model, the negative KL-divergence between model θ_q of query q and model θ_d of document d is calculated as follows:

$$\begin{aligned} -D(\theta_q||\theta_d) &= -\sum_{t\in V} p(t|\theta_q) \log \frac{p(t|\theta_q)}{p(t|\theta_d)} \\ &= \sum_{t\in V} p(t|\theta_q) \log p(t|\theta_d) - \sum_{t\in V} p(t|\theta_q) \log p(t|\theta_q) \\ &= \sum_{t\in V} p(t|\theta_q) \log p(t|\theta_d) + E(\theta_q), \end{aligned} \tag{6.1}$$

where $p(t|\theta_q)$ and $p(t|\theta_d)$ are the generative probabilities of a term t from the models θ_q and θ_d, V is the vocabulary of C, and $E(\theta_q)$ is the entropy of q. Given q,

$$E(\theta_q) = -\sum_{t\in V} p(t|\theta_q) \log p(t|\theta_q)$$

is a constant value that is independent of document d, so it will not affect the ranking of documents.

Define $tf(t,d)$ and $tf(t,q)$ as the frequencies of t in d and q, respectively. Normally, $p(t|\theta_q)$ is calculated with maximum likelihood estimation (MLE):

$$p(t|\theta_q) = \frac{tf(t,q)}{\sum_{t' \in q} tf(t',q)}. \tag{6.2}$$

Because $p(t|\theta_q) = 0$ for $t \notin q$, in the calculation of Equation 6.1, only the terms appearing in q are considered:

$$\begin{aligned} -D(\theta_q||\theta_d) &= \sum_{t \in V} p(t|\theta_q) \log p(t|\theta_d) + E(\theta_q) \\ &= \sum_{t \in q} p(t|\theta_q) \log p(t|\theta_d) + E(\theta_q). \end{aligned}$$

In the calculation of $p(t|\theta_d)$, several smoothing methods have been proposed to overcome the data sparseness problem of a language model constructed from one document (Zhai and Lafferty, 2001b). For example, $p(t|\theta_d)$ with Dirichlet-prior smoothing can be calculated as follows:

$$p(t|\theta_d) = \frac{tf(t,d) + \mu\ p(t|\theta_C)}{\sum_{t' \in V} tf(t',d) + \mu}, \tag{6.3}$$

where $\mu \geq 0$ is the prior parameter in the Dirichlet-prior smoothing method, and $p(t|\theta_C)$ is the probability of t in C, which is often calculated with MLE:

$$p(t|\theta_C) = \frac{\sum_{d' \in C} tf(t,d')}{\sum_{d' \in C} \sum_{t' \in V} tf(t',d')}.$$

6.1.2 Pseudo Relevance Feedback

Relevance feedback is widely used in IR to achieve better performance. It makes use of the documents retrieved for a given query to perform a new query by referring to the relevance judgment of these retrieved documents. One kind of relevance feedback methods is pseudo relevance feedback (PRF), which is also known as

blind relevance feedback. It assumes that the top k ranked documents in the initial retrieval are all relevant, so that the process of relevance judgment is automated. The process of PRF is constructed with two retrieval steps:

- In the first step, ranked documents are retrieved from C by a normal retrieval method with the original query q.
- In the second step, a number of terms are selected from the top k ranked documents D_q for query expansion, under the assumption that these k documents are relevant to the query. Then, the expanded query is used to retrieve the documents from C.

There are several methods to select expansion terms in the second step (Zhai and Lafferty, 2001a). For example, in Indri[3], the terms are first ranked by the following score:

$$v(t, D_q) = \sum_{d \in D_q} \log(\frac{tf(t,d)}{|d|} \times \frac{1}{p(t|\theta_C)}),$$

as in Ponte (1998). The terms with high frequency in D_q and low frequency in the text collection are selected as expansion terms. Define $p(q|\theta_d)$ as the probability of inducing q from a model of document d, which can be calculated as:

$$p(q|\theta_d) = e^{-D(\theta_q||\theta_d)}.$$

The top m terms T_q are selected with weights calculated based on the relevance model described in Lavrenko and Croft (2001):

$$w(t, D_q) = \sum_{d \in D_q} \left[\frac{tf(t,d)}{|d|} \times p(q|\theta_d) \times p(\theta_d) \right],$$

which calculates the sum of weighted probabilities of t in each document. Here, $p(\theta_d)$ is assumed to be uniform. After normalization, the probability of t in relevance

[3]http://lemurproject.org/indri/

model θ_q^r is calculated as follows:

$$p(t|\theta_q^r) = \frac{w(t, D_q)}{\sum_{t' \in T_q} w(t', D_q)}.$$

Finally, the relevance model is interpolated with the original query model:

$$p(t|\theta_q^{prf}) = \lambda \; p(t|\theta_q^r) + (1 - \lambda) p(t|\theta_q), \tag{6.4}$$

where parameter $\lambda \in [0, 1]$ controls the amount of feedback. The new query model θ_q^{prf} is used to replace the original one θ_q in Equation 6.1 in the second retrieval step of PRF.

6.1.2.1 Collection Enrichment

The documents retrieved from a text collection in the first retrieval step may not be sufficient to provide enough high quality feedback documents in the PRF method. Collection enrichment (CE) (Kwok and Chan, 1998) is a technique to improve the quality of the feedback documents by making use of an external target text collection X in addition to the original target C in the first step of PRF. The usage of X is meant to provide more relevant feedback documents such that we can generate high quality feedback query terms.

6.2 Word Sense Disambiguation

In this section, we introduce our method of disambiguating the queries and documents in IR. We first describe the construction of our WSD system. We then propose the method of assigning senses to terms in short queries.

6.2.1 Word Sense Disambiguation System

In Section 2.2.2, we highlighted that translations in another language can be used to disambiguate the meanings of words (Chan and Ng, 2005a; Zhong and Ng, 2009).

We construct our supervised WSD system directly from parallel corpora. Different from Chapter 5, we do not use the WordNet sense representation. Instead, the Chinese translation and the English morphological root of a word are directly used as the sense representation in this system to disambiguate the meanings for IR purpose. Therefore, no human effort is needed to annotate senses.

To generate the WSD training data, 7 parallel corpora were used, including *Chinese Treebank*, *FBIS Corpus*, *Hong Kong Hansards*, *Hong Kong Laws*, *Hong Kong News*, *Sinorama News Magazine*, and *Xinhua Newswire*. In total, there are 78 million English words and 111 million Chinese characters. These corpora are available from LDC and have already been aligned at sentence level. We tokenized English texts with the Penn Treebank Tokenizer, and performed word segmentation on Chinese texts using the Chinese word segmenter of (Low, Ng, and Guo, 2005). Then, word alignment was performed on the parallel corpora with the GIZA++ software (Och and Ney, 2003).

For each English morphological root e, the English sentences containing its occurrences were extracted from the word aligned output of GIZA++, as well as the corresponding translations of these occurrences. To minimize noisy word alignment result, translations with no Chinese character were deleted, and we further removed a translation when it only appears once, or its frequency is less than 10 and also less than 1% of the frequency of e. Finally, only the most frequent 10 translations were kept for efficiency consideration. The English part of the remaining occurrences were used as training data. Because multiple English words may have the same Chinese translation, to differentiate them, each Chinese translation is concatenated with the English morphological root to form a word sense. For example, "girl" and "woman" have the same Chinese translation "女子", but their corresponding senses, "girl_女子" for "girl" and "woman_女子" for "woman", are different.

In total, we extract training data for 63,921 English morphological roots.

32,153 of them have more than one sense (translation). Each ambiguous word has 5 senses and 740 training instances on average.

We use our supervised WSD system, *IMS*, introduced in Chapter 3, to train the WSD models. We choose MaxEnt as the machine learning algorithm, because it provides well-calibrated probability which will be helpful in the integration of word senses into the LM approach. The training process took about 183 hours in total using one CPU. Finally, the system can disambiguate a word by assigning probabilities to its different senses.

6.2.2 Estimating Sense Distributions for Query Terms

In IR, both query terms and terms in the text collection can be ambiguous. Hence, WSD is needed to disambiguate these ambiguous terms. In most cases, the documents in a text collection are full articles. Therefore, a WSD system has sufficient context to disambiguate the words in the documents. In contrast, queries are usually short, often with only two or three terms in a query. Short queries pose a challenge to WSD systems, since there is insufficient context to disambiguate a term in a short query.

One possible solution to this problem is to find some text fragments that contain a query term. Suppose we already have a basic IR method which does not require any sense information, such as the stem-based LM approach. Similar to the PRF method, assuming that the top k documents retrieved by the basic method are relevant to the query, these k documents can be used to represent a query q and considered as relevant fragments about q (Broder et al., 2007; Bendersky, Croft, and Smith, 2010; He and Wu, 2011). We propose a method to estimate the sense probabilities of each query term of q from these top k retrieved documents.

Suppose the words in all documents of the text collection have been disambiguated with a WSD system, and each word occurrence w in document d is

assigned a vector of senses, $S(w)$. Define the probability of assigning sense s to w in d as $p(w, s, d)$. Given a query q, suppose D_q is the set of top k documents retrieved by the basic IR method, with the probability score $p(q|\theta_d)$ assigned to $d \in D_q$. For a query term $t \in q$, define $O(t, d)$ as the set of word occurrences in d with the same stem form as t.

Given a query term $t \in q$
$S(t, q) = \{\}$
$sum = 0$
for each document $d \in D_q$
  for each word occurrence $w \in O(t, d)$
    for each sense $s \in S(w)$
      $S(t, q) = S(t, q) \cup \{s\}$
      $p(t, s, q) = p(t, s, q) + p(q|\theta_d)\ p(w, s, d)$
      $sum = sum + p(q|\theta_d)\ p(w, s, d)$
    end
  end
end
for each sense $s \in S(t, q)$
  $p(t, s, q) = p(t, s, q)/sum$
end
Return $S(t, q)$, with probability $p(t, s, q)$ for $s \in S(t, q)$

Figure 6.1: The process of generating senses for query terms

Figure 6.1 shows the pseudocode of calculating the sense distribution for a query term t in q with D_q, where $S(t, q)$ is the set of senses assigned to t and $p(t, s, q)$ is the probability of tagging t as sense s:

$$p(t, s, q) = \frac{\sum_{d \in D_q} \sum_{w \in O(t,d)} p(q|\theta_d) p(w, s, d)}{\sum_{s' \in S(t,q)} \sum_{d \in D_q} \sum_{w \in O(t,d)} p(q|\theta_d) p(w, s', d)}$$

Basically, we utilized the sense distribution of the words with the same stem form in D_q as a proxy to estimate the sense probabilities of a query term. The retrieval scores $p(q|\theta_d)$ are used to weight the information from the corresponding retrieved documents in D_q.

6.3 Incorporating Senses into Language Modeling Approaches

In this section, we propose to incorporate senses into the LM approach to IR by adjusting term frequencies. We also describe the integration of sense synonym relations into our model.

6.3.1 Incorporating Senses

With the method described in the last section, both the terms in queries and in documents have been sense tagged. The next problem is to incorporate sense information into the language modeling approach.

Suppose $p(t, s, q)$ is the probability of tagging a query term $t \in q$ as sense s, and $p(w, s, d)$ is the probability of tagging a word occurrence $w \in d$ as sense s. Given a query q and a document d in the text collection C, we want to re-estimate the language models by making use of sense information assigned to them.

Define the frequency of sense s in d as the sum of the probabilities of words in d tagged as s:

$$stf(s, d) = \sum_{w \in d} p(w, s, d).$$

If D is a set of documents, $stf(s, D)$ is the sum of the frequencies of s in all documents in D:

$$stf(s, D) = \sum_{d \in D} stf(s, d).$$

If S is a set of senses and $*$ is a wild card which could be either a document or a set of documents, $stf(S, *)$ is the sum of the frequencies of all senses in S:

$$stf(S, *) = \sum_{s \in S} stf(s, *).$$

For a term $t \in q$, with senses $S(t, q)$:$\{s_1, s_2, ..., s_n\}$, suppose the vector of probabilities assigned to the senses of t is V:$\{p(t, s_1, q), p(t, s_2, q), ..., p(t, s_n, q)\}$ and

the vector of frequencies of $S(t,q)$ in d is W:$\{stf(s_1,d), stf(s_2,d), ..., stf(s_n,d)\}$. The function $\cos(t,q,d)$ calculates the cosine similarity between vector V and vector W:

$$\cos(t,q,d) = \frac{\sum_{s\in S(t,q)} p(t,s,q)stf(s,d)}{\sqrt{\sum_{s\in S(t,q)} p(t,s,q)^2}\sqrt{\sum_{s\in S(t,q)} stf(s,d)^2}}.$$

Assume D_t is a set of documents in C which contain any sense in $S(t,q)$, we define a function:

$$\overline{cos}(t,q) = \frac{\sum_{d\in D_t} cos(t,q,d)}{|D_t|},$$

which calculates the mean of the sense cosine similarities. We define a function:

$$\Delta cos(t,q,d) = cos(t,q,d) - \overline{cos}(t,q),$$

which calculates the difference between $cos(t,q,d)$ and the corresponding mean value. $\Delta cos(t,q,d)$ measures the relative similarity between of q and d with regard to D_t.

Given a query q, we adjust the term frequency of query term t in d with sense information integrated as follows:

$$tf_{sen}(t,d) = tf(t,d) + sen(t,q,d), \tag{6.5}$$

where the function $sen(t,q,d)$ is a measure of t's sense information in d, which is defined as follows:

$$sen(t,q,d) = \alpha^{\Delta cos(t,q,d)} stf(S(t,q),d). \tag{6.6}$$

In $sen(t,q,d)$, the last item $stf(S(t,q),d)$ calculates the sum of the sense frequencies of senses of t in d, which represents the amount of t's sense information in d. A document d containing more senses of t will get a higher term frequency $tf_{sen}(t,d)$. The first item $\alpha^{\Delta cos(t,q,d)}$ is a weight of the sense information concerning the relative sense similarity $\Delta cos(t,q,d)$, where the parameter $\alpha \geq 1$ controls the impact of sense similarity. When $\Delta cos(t,q,d)$ is larger than zero, such that the sense similarity of d and q according to t is above the mean value, the weight for the sense

information is larger than 1; otherwise, it is less than 1. The more similar they are, the larger the weight value. Therefore, this function gives a higher weight to a document which has more senses of t with a similar sense distribution. For $t \notin q$, because the sense set $S(t,q)$ is empty, $stf(S(t,q),d)$ equals to zero and $tf_{sen}(t,d)$ is identical to $tf(t,d)$.

With sense incorporated, the term frequency is influenced by the sense information. Consequently, the estimation of probability of t in d becomes query specific:

$$p(t|\theta_d^{sen}) = \frac{tf_{sen}(t,d) + \mu\ p(t|\theta_C^{sen})}{\sum_{t' \in V} tf_{sen}(t',d) + \mu}, \tag{6.7}$$

where the probability of t in C is re-calculated as:

$$p(t|\theta_C^{sen}) = \frac{\sum_{d' \in C} tf_{sen}(t,d')}{\sum_{d' \in C} \sum_{t' \in V} tf_{sen}(t',d')}.$$

6.3.2 Expanding with Synonym Relations

Words usually have some semantic relations with others. Synonym relation is one of the semantic relations commonly used to attempt to improve IR performance. In this part, we further integrate the synonym relations of senses into the LM approach.

Suppose $R(s)$ is the set of senses having synonym relation with sense s. Define $S(q)$ as the union of all senses assigned to any term in query q, $S(q) = \bigcup_{t \in q} S(t,q)$. Define $R(s,q)$ as the set of senses which have synonym relation with sense s but not in $S(q)$, $R(s,q) = R(s) - S(q)$. We update the frequency of a query term t in d by integrating the synonym relations as follows:

$$tf_{syn}(t,d) = tf_{sen}(t,d) + syn(t,q,d), \tag{6.8}$$

where $syn(t,q,d)$ is a function measuring the synonym information of t in d:

$$syn(t,q,d) = \sum_{s \in S(t,q)} \beta(s,q) p(t,s,q) stf(R(s,q),d).$$

The function $syn(t, q, d)$ consists of three parts. The last item $stf(R(s, q), d)$ is the sum of the sense frequencies of $R(s, q)$ in d. A document containing more synonym senses of s will be assigned a larger value. Notice that the synonym senses already appearing in $S(q)$ are excluded in the calculation, because the information of these senses has been used in some other places in the retrieval function. The frequency of synonyms, $stf(R(s, q), d)$, is first weighted by $p(t, s, q)$. Therefore, the synonym senses of a sense $s \in S(t, q)$ will get a higher weight if the probability of tagging $t \in q$ as sense s is high; otherwise, the impact of synonym senses of s will be lower. We further weight the frequency of synonyms with a scaling function:

$$\beta(s, q) = \min(1, \frac{stf(s, C)}{stf(R(s, q), C)}).$$

$0 \leq \beta(s, q) \leq 1$. When $stf(s, C)$, the frequency of sense s in C, is less than $stf(R(s, q), C)$, the frequency of $R(s, q)$ in C, the function $\beta(s, q)$ scales down the impact of synonyms according to the ratio of these two frequencies. Otherwise, the value of the scaling function is one. The usage of this scaling function makes sure that the overall impact of the synonym senses is not greater than the original word senses.

Accordingly, we have the probability of t in d updated to:

$$p(t|\theta_d^{syn}) = \frac{tf_{syn}(t, d) + \mu p(t|\theta_C^{syn})}{\sum_{t' \in V} tf_{syn}(t', d) + \mu}, \tag{6.9}$$

and the probability of t in C is calculated as:

$$p(t|\theta_C^{syn}) = \frac{\sum_{d' \in C} tf_{syn}(t, d')}{\sum_{d' \in C} \sum_{t' \in V} tf_{syn}(t', d')}.$$

With this language model, the probability of a query term in a document is enlarged by the synonyms of its senses; The more synonym senses in a document, the higher the probability. Consequently, documents with more synonym senses of the query terms will get higher retrieval rankings.

Notice that the sense in our WSD system consists of two parts: a morphological root and a Chinese translation. The Chinese translation not only disambiguates the sense of the morphological root, but also provides clues of connections among different senses. We assume in our work that senses with the same Chinese translation are synonyms to generate a set of synonyms for each sense. For example, the following senses, *female*_女子, *girl*_女子, and *woman*_女子, share the same Chinese translation 女子. Under our assumption, each of them will have the other two as its synonym senses:

$$\begin{aligned} R(female\_女子) &= \{girl\_女子, woman\_女子\} \\ R(girl\_女子) &= \{female\_女子, woman\_女子\} \\ R(woman\_女子) &= \{female\_女子, girl\_女子\} \end{aligned}$$

With the synonym senses generated based on above assumption, we can utilize these synonym relations in the method proposed above.

6.4 Experiments

In this section, we evaluate and analyze the models proposed in Section 6.3 on standard TREC collections. We first describe our experimental settings and then discuss the experimental results.

6.4.1 Experimental Settings

We conduct experiments on the TREC collection. The text collection C includes the documents from TREC disk 4 and 5, minus the CR (Congressional Record) corpus, with 528,155 documents in total. In addition, the other documents in TREC disk 1 to 5 are used as the external text collection X for collection enrichment purpose. In total, we have 1.6 million documents in the union of the original text collection

and the external text collection.

Query Set	Topics	#qry	Avg	Rels
TREC6	301–350	50	2.58	4,290
TREC7	351–400	50	2.50	4,674
TREC8	401–450	50	2.46	4,728
RB03	601–650	50	3.00	1,658
RB04[4]	651–700	49	2.96	2,062

Table 6.1: Statistics of query sets

We use 50 queries from TREC6 Ad Hoc task as the development set, and evaluate on 50 queries from TREC7 Ad Hoc task, 50 queries from TREC8 Ad Hoc task, 50 queries from ROBUST 2003 (RB03), and 49 queries from ROBUST 2004 (RB04). In total, our test set includes 199 queries. We use the terms in the title field of TREC topics as queries. Table 6.1 shows the statistics of the five query sets. The first column lists the query topics, and the column *#qry* is the number of queries. The column *Avg* gives the average query length, and the column *Rels* is the total number of relevant documents.

We use the *Lemur* toolkit (Ogilvie and Callan, 2001) version 4.11 as the basic retrieval tool, and select the default unigram LM approach based on KL-divergence and Dirichlet-prior smoothing in Lemur as our basic retrieval approach. Stop words are removed from queries and documents using the standard INQUERY stop words list (Allan et al., 2000), and then the Porter stemmer is applied to perform stemming. The stem forms are finally used for indexing and retrieval.

We set the Dirichlet-prior smoothing parameter μ in Equation 6.3 to 400 by tuning on the TREC6 query set in a range of $\{100, 400, 700, 1000, 1500, 2000, 3000, 4000, 5000\}$. With this basic method, up to 10 top ranked documents D_q are retrieved for each query q from the extended text collection $C \cup X$, for the usage of performing PRF and generating query senses.

[4]Topic 672 is eliminated, since it has no relevant document.

For PRF, we follow the implementation of Indri's PRF method and further apply the CE technique as described in Section 6.1.2. The number of terms selected from D_q for expansion is tuned from $\{20, 25, 30, 35, 40\}$ and set to 25. The interpolation parameter λ in Equation 6.4 is set to 0.7 by tuning from $\{0.1, 0.2, ..., 0.9\}$. The CE-PRF method with this parameter setting is chosen as the baseline $Stem_{prf}$.

As shown in column *Avg* of Table 6.1, the queries in both development set and test set are short, with less than 3 query terms on average. To estimate the sense distributions for terms in these short queries, the method described in Section 6.2.2 is applied with D_q, the feedback documents in the PRF method, as the relevant text fragments. To disambiguate the documents in the text collection $C \cup X$, we employ the supervised WSD system described in Section 6.2.1. The disambiguation of the 1.6 million documents takes about 700 hours with one 2.83GHz CPU, and this process can be trivially parallelized. Besides the supervised WSD system, two WSD baseline methods, *MFS* and *Even*, are used for comparison. The method *MFS* tags the words with their most frequent senses, and the method *Even* assigns equal probabilities to all senses for each word.

We assign senses to the words in a query based on the documents sense-tagged using each WSD method. By applying the sense integration approach proposed in Section 6.3.1, we have three methods $Stem_{prf}$+*MFS*, $Stem_{prf}$+*Even*, and $Stem_{prf}$+*WSD*, for the MFS baseline, the Even baseline, and the supervised WSD system, respectively. We then tune the parameter α in Equation 6.6 on *TREC6* from 1 to 10 in increment of 1 for each WSD method. This parameter is set to 9, 6, and 7 for the method $Stem_{prf}$+*MFS*, $Stem_{prf}$+*Even*, and $Stem_{prf}$+*WSD*, respectively.

In Section 6.3.2, we assume that senses with the same Chinese translation are synonyms. Under this assumption, we generate a set of synonyms for each sense. We integrate synonym information into the methods $Stem_{prf}$+{*MFS*, *Even*,

WSD} with the approach proposed in Section 6.3.2.

6.4.2 Experimental Results

For evaluation, we use average precision (AP) as the metric to evaluate the performance on each query q:

$$AP(q) = \frac{\sum_{r=1}^{R} [p(r)rel(r)]}{relevance(q)},$$

where $relevance(q)$ is the number of documents relevant to q, R is the number of retrieved documents, r is the rank, $p(r)$ is the precision of the top r retrieved documents, and $rel(r)$ equals to 1 if the rth document is relevant, and 0 otherwise. Mean average precision (MAP) is a metric to evaluate the performance on a set of queries Q:

$$MAP(Q) = \frac{\sum_{q \in Q} AP(q)}{|Q|},$$

where $|Q|$ is the number of queries in Q.

We retrieve the top-ranked 1,000 documents for each query, and use the MAP score as the main evaluation metric. In Table 6.2, the first four columns are the MAP scores of various methods on the TREC7, TREC8, RB03, and RB04 query set, respectively. The column *Comb* shows the results on the union of the four test query sets. The first three rows list the results of the top three systems that participated in the corresponding tasks. The row $Stem_{prf}$ shows the performance of our baseline method, the stem-based CE-PRF method. The column *Impr* calculates the percentage improvement of each method over the baseline $Stem_{prf}$ in column *Comb*. The last column *#ret-rel* lists the total numbers of relevant documents retrieved by different methods.

Comparing to the top participating systems, the performance of our baseline method $Stem_{prf}$ is relatively strong. It outperforms all the participating systems in TREC7 and achieves competitive performance on the other three query sets.

Method	TREC7	TERC8	RB03	RB04	Comb	Impr	#ret-rel
Top 1	0.2530	0.3063	0.3704	**0.4019**	-	-	-
Top 2	0.2488	0.2876	0.3065	0.4008	-	-	-
Top 3	0.2427	0.2853	0.3037	0.3514	-	-	-
Stem_{prf} (Baseline)	0.2634	0.2944	0.3586	0.3781	0.3234	-	9248
Stem_{prf}+MFS	0.2655	0.2971	$0.3626^{\dagger}$	0.3802	$0.3261^{\dagger}$	0.84%	9281
Stem_{prf}+Even	0.2655	0.2972	$0.3623^{\dagger}$	0.3814	$0.3263^{\ddagger}$	0.91%	9284
Stem_{prf}+WSD	$0.2679^{\ddagger}$	$0.2986^{\dagger}$	$0.3649^{\ddagger}$	0.3842	$0.3286^{\ddagger}$	1.63%	9332
Stem_{prf}+MFS+Syn	$0.2756^{\ddagger}$	$0.3034^{\dagger}$	$0.3649^{\dagger}$	0.3859	$0.3322^{\ddagger}$	2.73%	9418
Stem_{prf}+Even+Syn	$0.2713^{\dagger}$	$0.3061^{\ddagger}$	$0.3657^{\ddagger}$	$0.3859^{\dagger}$	$0.3320^{\ddagger}$	2.67%	9445
Stem_{prf}+WSD+Syn	$\mathbf{0.2762}^{\ddagger}$	$\mathbf{0.3126}^{\ddagger}$	$\mathbf{0.3735}^{\ddagger}$	$0.3891^{\dagger}$	$0.3376^{\ddagger}$	4.39%	9538

Table 6.2: Results on the test sets in MAP score. The first three rows show the results of the top participating systems, the next row shows the performance of the baseline method, and the remaining rows are the results of our method with different settings. Single dagger (†) and double dagger (‡) indicate statistically significant improvement over $Stem_{prf}$ at the 95% and 99% confidence level with a two-tailed paired t-test, respectively. The best results are highlighted in bold.

The rows $Stem_{prf}+\{MFS, Even, WSD\}$ are the results of incorporating word senses. Comparing to the baseline method, all methods with sense integrated achieve consistent improvements on all query sets. The usage of the supervised WSD method outperforms the other two WSD baselines, and it achieves statistically significant improvements over $Stem_{prf}$ on TREC7, TREC8, and RB03.

The integration of senses into the baseline method achieves two effects. First, using morphological roots, words with different inflectional forms but the same morphological root can be normalized in this sense representation. Thus, documents containing irregular inflections are retrieved when senses are integrated. For example, in topic 326 {*ferry sinkings*}, the stem form of *sinkings* is *sink*. As *sink* is an irregular verb, the usage of senses improves the retrieval recall by retrieving documents containing the inflection forms *sunk*, *sank*, and *sunken*.

Second, the senses output by our supervised WSD system help to identify the meanings of query terms. Take topic 357 {*territorial waters dispute*} for example. The stem form of *waters* is *water* and its appropriate sense in this query should be *water*_水域 (body of water) instead of the most frequent sense *water*_水 (H_2O). In $Stem_{prf}+WSD$, we correctly identify the minority sense for this query term:

sense	probability
*water*_水域 (body of water)	0.700
*water*_水 (H_2O)	0.047
*water*_供水 (provide with water)	0.025
...	...

Therefore, the documents in which water is tagged as *water*_水域 will be ranked higher using the method $Stem_{prf}+WSD$.

In another example topic 425 {*counterfeiting money*}, the stem form of *counterfeiting* is *counterfeit*. Although the most frequent sense *counterfeit*_冒牌 (not genuine) is not wrong, another sense *counterfeit*_伪币 (forged money) is more accu-

rate for this query term. The Chinese translation in the latter sense represents the meaning of the phrase in the original query. $Stem_{prf}+WSD$ outperforms the other two methods on this query by assigning the highest probability for this sense:

sense	probability
*counterfeit*_伪币 (forged money)	0.204
*counterfeit*_伪造 (forge, fake)	0.116
*counterfeit*_冒牌 (not genuine)	0.104
...	...

Overall, the performance of $Stem_{prf}+WSD$ is better than $Stem_{prf}+\{MFS, Even\}$ on 121 queries and 119 queries, respectively. The *t-test* at the confidence level of 99% indicates that the improvements are statistically significant.

The results of expanding with synonym relations in the above three methods are shown in the last three rows, $Stem_{prf}+\{MFS, Even, WSD\}$+Syn. The integration of synonym relations further improves the performance no matter what kind of sense tagging method is applied. The improvement varies with different methods on different query sets. As shown in the last column of Table 6.2, the number of relevant documents retrieved increases for each method. $Stem_{prf}+Even+Syn$ retrieves more relevant documents than $Stem_{prf}+MFS+Syn$, because the former method expands more senses. Overall, $Stem_{prf}+WSD+Syn$ achieves larger improvements than the other two methods. It shows that the WSD technique can help choose the appropriate senses for synonym expansion. For example, in topic 648 {*family leave law*}, the senses assigned to query term *leave* with supervised WSD system are as follows:

sense	probability
*leave*_假期 (leave of absence)	0.371
*leave*_离开 (go away)	0.198
...	...

The sense *leave*_假期 is assigned the highest probability instead of the dominant sense *leave*_离开. Thus, documents containing the synonym senses *holiday*_假期 and *vacation*_假期[5] are ranked higher in the method $Stem_{prf}+WSD+Syn$.

Among the different settings, $Stem_{prf}+WSD+Syn$ achieves the best performance. Its improvement over the baseline method is statistically significant at the 95% confidence level on RB04 and at the 99% confidence level on the other three query sets, with an overall improvement of 4.39%. It beats the best participating systems on three out of four query sets, including *TREC7, TREC8*, and *RB03*. On *RB04*, the top two systems are the results of the same participant with different configurations. One advantange of these two systems over our system is that they used lots of Web resources, such as search engines, to improve the performance.

The average speed of retrieval per query is 2 seconds for retrieval with senses, and 11 seconds for retrieval with senses and synonyms, using a computer with 2.83GHz CPU and 16GB memory. This process can be sped up by performing the retrieval of multiple query terms as well as their senses and synonyms in parallel.

In our method, the senses of words are translations constructed from parallel texts. We choose the English-Chinese language pair, since the two languages are from distantly related language families, such that the Chinese translations can better distinguish the meanings of English words. Instead of relying on a predefined list of senses from WordNet or other dictionaries, the use of translations as sense representation partially solves the granularity problem of WSD. We integrate the predicted probabilities of senses into the IR model. The "softer" way of integrating the predicted senses reduces the negative effects of WSD errors. Similar to the successful application of WSD to MT, WSD is used not as a black-box module in our method. These results suggest that the sense discriminators and the applications of WSD techniques should be task dependent.

[5]In WordNet, *leave*, *holiday*, and *vacation* have the same ancestor *time off* (a time period when you are not required to work).

6.5 Summary

This chapter reports the successful application of WSD to IR. We proposed a method for assigning senses to terms in short queries, and also described an approach to integrate senses into an LM approach for IR. In experiments on four query sets of TREC, we compared the performance of a supervised WSD method with two baseline WSD methods. Our experimental results showed that the incorporation of senses improved a state-of-the-art baseline, a stem-based LM approach with PRF. The performance of the supervised WSD method is better than the other two baseline WSD methods. We also proposed a method to further integrate synonym relations into the LM approach. With the integration of synonym relations, our best system with supervised WSD achieved an improvement of 4.39% over the baseline method, and it outperformed the best participating systems on three out of four query sets.

Chapter 7

Conclusion

The topic studied in this thesis is word sense disambiguation (WSD). We have investigated domain adaptation in WSD, the lack of training data for WSD, and the application of WSD in IR.

To promote research of supervised WSD approaches and the application of WSD, we develop and release an open source supervised WSD system, IMS. The flexible framework of IMS allows users to integrate different preprocessing tools, additional features, and different classifiers. With the default implementation of features and classifiers, as well as the classification models in the package, IMS achieves state-of-the-art accuracies on several SensEval/SemEval English lexical-sample tasks and all-words tasks. Therefore, IMS not only provides a platform for supervised WSD research, but also provides an all-words WSD component with good performance for other applications.

We examine the domain adaptation problem in WSD with SemCor and OntoNotes datasets. Using OntoNotes as the target domain data and SemCor as the source domain data, we show that the performance of a supervised WSD system is domain dependent. A supervised WSD system, which achieves a high accuracy with target domain training data, suffers a drop of more than 10% in

accuracy when it is trained on source domain data. To overcome this problem, we apply the feature augmentation technique to WSD, and propose a method which combines this technique and active learning to reduce the annotation effort required for adaptation. Through the experiments, we show that our method is able to greatly reduce the annotation effort required for domain adaptation and obtain a substantial improvement in accuracy. Our study suggests that, when applying a previously trained WSD system to a different domain, it is worth the effort to use our method to annotate a small number of examples on the most frequent word types from the new domain.

Next, we propose a method to extract sense-annotated examples from parallel corpora without extra human effort. To deal with the bottleneck of lack of sense-annotated data for WSD, we extend the work of Chan and Ng (2005a) by automatically selecting Chinese translations for English senses using bilingual dictionaries and bilingual corpora. Evaluation on OntoNotes data shows that the training examples gathered with our approach are of high quality, and result in statistically significant improvement in WSD accuracy. The major advantage of our approach is that the process of extracting sense-annotated examples from parallel corpora becomes completely unsupervised.

Finally, we propose to integrate senses into an IR system based on the LM approach, and with further integration of sense synonym relations. Our evaluation on several TREC query sets shows that adding WSD to IR achieves statistically significant improvements over a state-of-the-art IR baseline system.

7.1 Future Work

We have successfully applied the feature augmentation technique to the domain adaptation problem in WSD. Our adaptation approach is supervised in that we make use of some sense-annotated examples from the target domain. A potential

future direction is to investigate the use of a semi-supervised extension of the feature augmentation technique, which makes use of both labeled and unlabeled target domain data (Daumé III, Kumar, and Saha, 2010). Such an extension may further reduce the human effort needed for adaptation.

The hypothesis of our method of automatic extraction of training data from parallel corpora is that the senses of a word often have distinct translations in a second language. So far, we have shown that this method can gather high quality training data for English from English-Chinese parallel corpora. Because the hypothesis is also applicable to other languages, our method can be applied to gather training data from parallel corpora for other languages. In particular, the existing English-Chinese parallel corpora can be used for Chinese WSD.

Similar to IR, the question answering task needs WSD to disambiguate the terms in questions and documents. Besides identifying the relevant documents of a given question, WSD can also help to disambiguate the information to be extracted. Therefore, another potential direction is to improve question answering systems with an appropriate integration of WSD as a component.

References

Agirre, Eneko, Xabier Arregi, and Arantxa Otegi. 2010. Document expansion based on WordNet for robust IR. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 9–17.

Agirre, Eneko and Philip Glenny Edmonds. 2006. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science + Business Media.

Agirre, Eneko and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12.

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

Akkaya, Cem, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 190–199.

Allan, James, Margaret E. Connell, W. Bruce Croft, Fang-Fang Feng, David Fisher, and Xiaoyan Li. 2000. INQUERY and TREC-9. In *Proceedings of the 9th Text REtrieval Conference (TREC)*, pages 551–562.

Ando, Rie Kubota. 2006. Applying alternating structure optimization to word sense disambiguation. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, pages 77–84.

Balamurali, A R, Aditya Joshi, and Pushpak Bhattacharyya. 2011. Harnessing WordNet senses for supervised sentiment classification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1091.

Banerjee, Satanjeev and Ted Pedersen. 2002. 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)*, pages 136–145.

Bendersky, Michael, W. Bruce Croft, and David A. Smith. 2010. Structural annotation of search queries using pseudo-relevance feedback. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1537–1540.

Bloehdorn, Stephan and Andreas Hotho. 2004. Boosting for text classification with semantic features. In *Proceedings of the 6th International Conference on Knowledge Discovery on the Web: Advances in Web Mining and Web Usage Analysis*, pages 70–87.

Brin, S. and L. Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

Broder, Andrei Z., Marcus Fontoura, Evgeniy Gabrilovich, Amruta Joshi, Vanja Josifovski, and Tong Zhang. 2007. Robust classification of rare queries using web knowledge. In *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 231–238.

Brown, Peter F., Stephen A. Della Pietra, Vincent. J. Della Pietra, and Robert L. Mercer. 1991. Word-sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 264–270.

Cabezas, Clara and Philip Resnik. 2005. Using WSD techniques for lexical selection in statistical machine translation. Technical report, University of Maryland.

Cao, Guihong, Jian-Yun Nie, and Jing Bai. 2005. Integrating word relationships into language models. In *Proceedings of the 28th International ACM SIGIR*

Conference on Research and Development in Information Retrieval (SIGIR), pages 298–305.

Carpuat, Marine, Weifeng Su, and Dekai Wu. 2004. Augmenting ensemble classification for word sense disambiguation with a kernel PCA model. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SensEval-3)*, pages 88–92.

Carpuat, Marine and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 387–394.

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

Carreras, Xavier and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 152–164.

Chan, Yee Seng and Hwee Tou Ng. 2005a. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 1037–1042.

Chan, Yee Seng and Hwee Tou Ng. 2005b. Word sense disambiguation with distribution estimation. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1010–1015.

Chan, Yee Seng and Hwee Tou Ng. 2006. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 89–96.

Chan, Yee Seng and Hwee Tou Ng. 2007. Domain adaptation with active learning

for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 49–56.

Chan, Yee Seng, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 33–40.

Chan, Yee Seng, Hwee Tou Ng, and Zhi Zhong. 2007. NUS-PT: Exploiting parallel texts for word sense disambiguation in the English all-words tasks. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 253–256.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–180.

Chen, Jingying, Andrew Schein, Lyle Ungar, and Martha Palmer. 2006. An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 120–127.

Chiang, David, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 218–226.

Chklovski, Timothy and Rada Mihalcea. 2002. Building a sense tagged corpus with open mind word expert. In *Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 116–122.

Collins, Michael. 1999. *Head-Driven Statistical Model for Natural Language Parsing*. PhD dissertation, University of Pennsylvania.

Dang, Hoa Trang and Martha Palmer. 2005. The role of semantic roles in disambiguating verb senses. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 42–49.

Daumé III, Hal. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 256–263.

Daumé III, Hal, Abhishek Kumar, and Avishek Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59.

Daumé III, Hal and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC)*, pages 449–454.

Decadt, Bart, Veronique Hoste, and Walter Daelemans. 2004. GAMBL, genetic algorithm optimization of memory-based WSD. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SensEval-3)*, pages 108–112.

Diab, Mona and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 255–262.

Escudero, Gerard, Lluís Màrquez, and German Riagu. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 172–180.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Fang, Hui. 2008. A re-examination of query expansion using lexical resources. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 139–147.

Florian, Radu, Silviu Cucerzan, Charles Schafer, and David Yarowsky. 2002. Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4):327–341.

Fujii, Atsushi, Takenobu Tokunaga, Kentaro Inui, and Hozumi Tanaka. 1998. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597.

Giménez, Jesús and Lluís Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 159–166.

Gonzalo, Julio, Anselmo Penas, and Felisa Verdejo. 1999. Lexical ambiguity and information retrieval revisited. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 195–202.

Gonzalo, Julio, Felisa Verdejo, Irina Chugur, and Juan Cigarrin. 1998. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the ACL Workshop on Usage of WordNet for NLP*, pages 38–44.

He, Daqing and Dan Wu. 2011. Enhancing query translation with relevance feedback in translingual information retrieval. *Information Processing & Management*, 47(1):1–17.

Hearst, Marti A. 1991. Noun homograph disambiguation using local context in

large corpora. In *Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary*, pages 1–22.

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

Huang, Chu-Ren, Ru-Yng Chang, and Hsiang-Pin Lee. 2004. Sinica BOW (Bilingual Ontological WordNet): Integration of bilingual WordNet and SUMO. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1553–1556.

Ide, Nancy and Jean Veronis. 1998. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40.

Jeh, Glen and Jennifer Widom. 2003. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web (WWW)*, pages 271–279.

Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19–33.

Kehagias, Athanasios, Vassilios Petridis, Vassilis G. Kaburlasos, and Pavlina Fragkou. 2003. A comparison of word- and sense-based text categorization using several classification algorithms. *Journal of Intelligent Information Systems*, 21(3):227–247.

Kilgarriff, Adam. 2001. English lexical sample task description. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SensEval-2)*, pages 17–20.

Kilgarriff, Adam and Joseph Rosenzweig. 2000. Framework and results for English

SensEval. *Computers and the Humanities: Special Issue on SensEval*, 34(1-2):15–48.

Kim, Sang-Bum, Hee-Cheol Seo, and Hae-Chang Rim. 2004. Information retrieval using word senses: root sense tagging approach. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 258–265.

Klein, Dan, Kristina Toutanova, H. Tolga Ilhan, Sepandar D. Kamvar, and Christopher D. Manning. 2002. Combining heterogeneous classifiers for word-sense disambiguation. In *Proceedings of the ACL Workshop on Word Sense Disambiguation*, pages 74–80.

Kohomban, Upali S. and Wee Sun Lee. 2005. Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 34–41.

Krovetz, Robert and W. Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141.

Kwok, Kui-Lam and Margaret Chan. 1998. Improving two-stage ad-hoc retrieval for short queries. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 250–256.

Lafferty, John and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 111–119.

Lavrenko, Victor and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 120–127.

Lee, Yoong Keok and Hwee Tou Ng. 2002. An empirical evaluation of knowledge

sources and learning algorithms for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 41–48.

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

Lewis, David D. and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 3–12.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL)*, pages 64–71.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL-COLING)*, pages 768–774.

Liu, Shuang, Fang Liu, Clement Yu, and Weiyi Meng. 2004. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 266–272.

Liu, Shuang, Clement Yu, and Weiyi Meng. 2005. Word sense disambiguation in queries. In *Proceedings of the 14th ACM Conference on Information and Knowledge Management (CIKM)*, pages 525–532.

Low, Jin Kiat, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 161–164.

Magnini, Bernardo, Danilo Giampiccolo, and Alessandro Vallin. 2004. The Italian lexical sample task at Senseval-3. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SensEval-3)*, pages 17–20.

Manandhar, Suresh, Ioannis P. Klapaftis, Dmitriy Dligach, and Sameer S. Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 63–68.

Marcus, Mitchell, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Màrquez, Lluis, Mariona Taulé, Antonia Martí, Núria Artigas, Mar García, Francis Real, and Dani Ferrés. 2004. Senseval-3: The Spanish lexical sample task. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SensEval-3)*, pages 21–24.

Martinez, David and Eneko Agirre. 2000. One sense per collocation and genre/topic variations. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 207–215.

McCarthy, Diana, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 279–286.

Mihalcea, Rada. 2002. Bootstrapping large sense tagged corpora. In *Proceedings*

of the 3rd International Conference on Languages Resources and Evaluation (LREC), pages 1407–1411.

Mihalcea, Rada. 2004. Co-training and self-training for word sense disambiguation. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, pages 33–40.

Mihalcea, Rada. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 411–418.

Mihalcea, Rada, Timothy Chklovski, and Adam Kilgarriff. 2004. The SensEval-3 English lexical sample task. In *Proceedings of the Third International Workshop on Evaluating Word Sense Disambiguation Systems (SensEval-3)*, pages 25–28.

Mihalcea, Rada and Andras Csomai. 2005. SenseLearner: Word sense disambiguation for all words in unrestricted text. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL) Interactive Poster and Demonstration Sessions*, pages 53–56.

Mihalcea, Rada and Dan Moldovan. 2001. Pattern learning and active feature selection for word sense disambiguation. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SensEval-2)*, pages 127–130.

Mihalcea, Rada, Paul Tarau, and Elizabeth Figa. 2004. Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 1126–1132.

Miller, George A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Miller, George A., Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 240–243.

Navigli, Roberto and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1683–1688.

Navigli, Roberto and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

Navigli, Roberto, Kenneth Litkowski, and Orin Hargraves. 2007. SemEval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35.

Navigli, Roberto and Paola Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1075–1086.

Ng, Hwee Tou. 1997a. Exemplar-based word sense disambiguation: Some recent improvements. In *Proceedings of the 1997 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 208–213.

Ng, Hwee Tou. 1997b. Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How*, pages 1–7.

Ng, Hwee Tou and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 40–47.

Ng, Hwee Tou, Bin Wang, and Yee Seng Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 455–462.

Niu, Zheng-Yu, Dong-Hong Ji, and Chew Lim Tan. 2004. Optimizing feature set for chinese word sense disambiguation. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SensEval-3)*, pages 191–194.

Niu, Zheng-Yu, Dong-Hong Ji, and Chew Lim Tan. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 395–402.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932.

Och, Franz Josef and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Ogilvie, Paul and Jamie Callan. 2001. Experiments using the Lemur toolkit. In *Proceedings of the 10th Text REtrieval Conference (TREC)*, pages 103–108.

Palmer, Martha, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Pro-*

ceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SensEval-2), pages 21–24.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Pedersen, Ted. 2000. A simple approach to building ensembles of naïve Bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL)*, pages 63–69.

Pedersen, Ted, Satanjeev Banerjee, and Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. Research report, University of Minnesota Supercomputing Institute.

Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity – measuring the relatedness of concepts. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004): Demonstration Papers*, pages 38–41.

Pham, Thanh Phong, Hwee Tou Ng, and Wee Sun Lee. 2005. Word sense disambiguation with semi-supervised learning. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 1093–1998.

Ponte, Jay M. 1998. *A Language Modeling Approach to Information Retrieval*. Ph.D. thesis, Department of Computer Science, University of Massachusetts.

Ponte, Jay M. and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 275–281.

Ponzetto, Simone Paolo and Roberto Navigli. 2010. Knowledge-rich word sense

disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1522–1531.

Pradhan, Sameer, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92.

Rada, R., H. Mili, E. Bicknell, and M. Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30.

Resnik, Philip. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Resnik, Philip and David Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How*, pages 79–86.

Resnik, Philip and David Yarowsky. 2000. Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133.

Sanderson, Mark. 1994. Word sense disambiguation and information retrieval. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 142–151.

Sanderson, Mark. 2000. Retrieving with good sense. *Information Retrieval*, 2(1):49–69.

Sanderson, Mark. 2008. Ambiguous queries: test collections need more sense. In

Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 499–506.

Schütze, Hinrich. 1992. Dimensions of meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 787–796.

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

Schütze, Hinrich and Jan O. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Sinha, Ravi and Rada Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the First IEEE International Conference on Semantic Computing*, pages 363–369.

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

Stokoe, Christopher, Michael P. Oakes, and John Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 159–166.

Tratz, Stephen, Antonio Sanfilippo, Michelle Gregory, Alan Chappell, Christian Posse, and Paul Whitney. 2007. PNNL: A supervised maximum entropy approach to word sense disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 264–267.

Vasilescu, Florentina, Philippe Langlais, and Guy Lapalme. 2004. Evaluating variants of the Lesk approach for disambiguating words. In *Proceedings of*

the Fifth Conference on Language Resources and Evaluation (LREC), pages 633–636.

Veenstra, Jorn, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and Jakub Zavrel. 2000. Memory based word sense disambiguation. *Computers and the Humanities*, 34(1-2):171–177.

Vickrey, David, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 771–778.

Voorhees, Ellen M. 1993. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 171–180.

Voorhees, Ellen M. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 61–69.

Wang, Xinglong and Joh Carroll. 2005. Word sense disambiguation using sense examples automatically acquired from a second language. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 547–554.

Weaver, Warren. 1955. Translation. In William N. Locke and A. Donald Booth, editors, *Machine Translation of Languages*. Technology Press of MIT, Cambridge, MA, and John Wiley & Sons, New York, NY, pages 15–23.

Wiebe, Janyce and Rada Mihalcea. 2006. Word sense and subjectivity. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1065–1072.

Witten, Ian H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition.

Yarowsky, David. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 88–95.

Yarowsky, David. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(1–2):179–186.

Yarowsky, David, Radu Florian, Siviu Cucerzan, and Charles Schafer. 2001. The Johns Hopkins SensEval-2 system description. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SensEval-2)*, pages 163–166.

Zhai, Chengxiang and John Lafferty. 2001a. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th ACM Conference on Information and Knowledge Management (CIKM)*, pages 403–410.

Zhai, Chengxiang and John Lafferty. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 334–342.

Zhong, Zhi and Hwee Tou Ng. 2009. Word sense disambiguation for all words without hard labor. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1616–1621.

Zhong, Zhi and Hwee Tou Ng. 2010. It Makes Sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 78–83.

Zhong, Zhi and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 273–282.

Zhong, Zhi, Hwee Tou Ng, and Yee Seng Chan. 2008. Word sense disambiguation using OntoNotes: An empirical study. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1002–1010.

Zhu, Jingbo and Eduard Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 783–790.