Closing the Gap in WSD: Supervised Results with Unsupervised Methods

Samuel Brody

Doctor of Philosophy

Institute for Communicating and Collaborative Systems
School of Informatics
University of Edinburgh

2009

Abstract

Word-Sense Disambiguation (WSD), holds promise for many NLP applications requiring broad-coverage language understanding, such as summarization (Barzilay and Elhadad, 1997) and question answering (Ramakrishnan et al., 2003). Recent studies have also shown that WSD can benefit machine translation (Vickrey et al., 2005) and information retrieval (Stokoe, 2005). Much work has focused on the computational treatment of sense ambiguity, primarily using data-driven methods. The most accurate WSD systems to date are supervised and rely on the availability of sense-labeled training data. This restriction poses a significant barrier to widespread use of WSD in practice, since such data is extremely expensive to acquire for new languages and domains.

Unsupervised WSD holds the key to enable such application, as it does not require sense-labeled data. However, unsupervised methods fall far behind supervised ones in terms of accuracy and ease of use. In this thesis we explore the reasons for this, and present solutions to remedy this situation. We hypothesize that one of the main problems with unsupervised WSD is its lack of a standard formulation and general purpose tools common to supervised methods. As a first step, we examine existing approaches to unsupervised WSD, with the aim of detecting independent principles that can be utilized in a general framework. We investigate ways of leveraging the diversity of existing methods, using ensembles, a common tool in the supervised learning framework. This approach allows us to achieve accuracy beyond that of the individual methods, without need for extensive modification of the underlying systems.

Our examination of existing unsupervised approaches highlights the importance of using the predominant sense in case of uncertainty, and the effectiveness of statistical similarity methods as a tool for WSD. However, it also serves to emphasize the need for a way to merge and combine learning elements, and the potential of a supervised-style approach to the problem. Relying on existing methods does not take full advantage of the insights gained from the supervised framework.

We therefore present an unsupervised WSD system which circumvents the question of actual disambiguation method, which is the main source of discrepancy in unsupervised WSD, and deals directly with the data. Our method uses statistical and semantic similarity measures to produce labeled training data in a completely unsupervised fashion. This allows the training and use of any standard supervised classifier for the actual disambiguation. Classifiers trained with our method significantly outperform those using other methods of data generation, and represent a big step in bridging the accuracy

gap between supervised and unsupervised methods.

Finally, we address a major drawback of classical unsupervised systems – their reliance on a fixed sense inventory and lexical resources. This dependence represents a substantial setback for unsupervised methods in cases where such resources are unavailable. Unfortunately, these are exactly the areas in which unsupervised methods are most needed. Unsupervised sense-discrimination, which does not share those restrictions, presents a promising solution to the problem. We therefore develop an unsupervised sense discrimination system. We base our system on a well-studied probabilistic generative model, Latent Dirichlet Allocation (Blei et al., 2003), which has many of the advantages of supervised frameworks. The model's probabilistic nature lends itself to easy combination and extension, and its generative aspect is well suited to linguistic tasks. Our model achieves state-of-the-art performance on the unsupervised sense induction task, while remaining independent of any fixed sense inventory, and thus represents a fully unsupervised, general purpose, WSD tool.

Acknowledgements

First and foremost, my thanks go to Mirella Lapata, who is all one could wish for in a supervisor. Without her exceptional ability to encompass everything from high-level comments to missing commas, the quality of my work would have suffered greatly. It has been a pleasure to have her as my supervisor.

I am indebted to Diana McCarthy, John Carroll and Rob Koeling for their help, comments and suggestions in the process of our collaboration. I am grateful to Roberto Navigli for his friendly assistance and for providing the results of his SSI algorithm for our ensemble experiments, even at short notice. I wish to thank David Talbot for his insightful suggestion which provided the basis for much of my work on unsupervised data creation. I gratefully acknowledge the support of EPSRC (Brody and Lapata; grant EP/C538447/1), which made this thesis possible.

I would like to thanks my family for providing support and encouragement throughout. Special thanks to my grandmother, Rose Brody, for her constant concern for my wellbeing and steadfast belief in my abilities.

I am very grateful to my friends in Israel, especially Aviv Hurvitz, Yifat Monnickendam, Sarai Sheinvald and Robby Lampert, who kept in touch, took an active interest in my welfare, and provided a sympathetic ear when needed. I would also like to express my thanks to all my friends at the University of Edinburgh for making my years there a very pleasant experience. In this regard, Markus Becker, Chris Callison-Burch, Ben Hachey and Anna Pakarinen, Trevor Cohn, Josh Schroeder and Jackie Bedoya, and Sharon Givon deserve special mention. It is my hope that, despite geographical distance, we will remain close.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Samuel Brody*)

In memory of my mother, Ziporah Brody, who valued academic achievement, and knew when to put it aside in favor of more important things.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Ford: "You'd better be prepared for the jump into hyperspace. It's unpleasantly like being drunk."
> Arthur: "What's so unpleasant about being drunk?"
> Ford: "You ask a glass of water."
>
> ---
>
> *Hitchhiker's Guide to the Galaxy*
> Douglas Adams

Ambiguity is a widespread phenomenon throughout human perception and language, and a fundamental issue when attempting to understand human cognition. It appears in various forms in natural language, and is at the root of many basic problems in the area of computational linguistics, such as anaphora resolution, prepositional attachment, part-of-speech tagging and more. In this thesis we focus on the problem of word-sense disambiguation (WSD), the task of identifying the intended meanings (senses) of words in context. WSD is one of the oldest problems in computational linguistics, first formulated as a computational task in the 1940s (Weaver, 1949/1955), and is still a very active field of research. The most recent workshop for evaluation of WSD (Agirre et al., 2007), was the biggest ever, containing eighteen tasks with over one hundred participating teams. A famous example of the difficulty of the task was given by Bar-Hillel (1960). He claimed that no existing or imaginable computer program would be able to determine that the correct meaning of the word *pen* in the passage below is *enclosure*, rather than *writing-implement*.

> Little John was looking for his toy box. Finally he found it. The box was in the *pen*. John was very happy.

Ironically, as pointed out by Yarowsky (2000), this same example can be used to demonstrate the utility of automatic disambiguation methods using statistical mea-

sures, since it is very rare to refer to the contents of a writing implement, whereas this is much more common for an enclosure.

WSD is of interest and importance for several reasons. From a research perspective, it is an excellent representative case of ambiguity in language in general, embodying the correspondence between senses and meanings, while being self contained and well-defined. It is also of interest in its own right, since it has many unique characteristics, touching both semantics and syntax, and being strongly connected to many other areas of linguistic knowledge (as evidenced by the many and diverse approaches used to address the problem throughout the years, see Section 3.2). WSD has an additional layer of complexity, not found in other types of linguistic ambiguity, due to the fact that the list of potential senses varies from word to word and, in fact, may differ even for an individual word, as a result of change in domain or shifts in language usage over time. From a pragmatic perspective, WSD is important since it holds promise for many natural-language applications requiring broad-coverage language understanding. Examples include summarization, information retrieval and machine translation. For instance, when used as part of a machine translation system, WSD can greatly reduce the computational complexity and running time required to produce an accurate translation by detecting the correct sense of the source words, and removing from the search space translations pertaining to the other senses. For information retrieval, a more sophisticated indexing, where word-senses are used instead of ambiguous words, could make it easier for the user to find the information they wanted. Is would also reduce the number of entries for each key, thus making retrieval faster. To enable such an indexing, an accurate wide-coverage WSD method is necessary in order to disambiguate all the words in the document database. Recent studies show that correctly applied WSD benefits both machine translation (Vickrey et al., 2005) and information retrieval (Stokoe, 2005).

Given the potential of WSD for many NLP tasks, much work has focused on the computational treatment of sense ambiguity, primarily using data-driven methods. Most accurate WSD systems to date are supervised (see Pradhan et al. 2007). Such systems use classification algorithms trained on occurrences of ambiguous words which were manually annotated with the appropriate sense given the context. The classifiers automatically learn disambiguation cues from these hand-labeled examples.

Although supervised methods typically achieve better performance than their unsupervised alternatives, their applicability is limited to those words for which sense labeled data exists, and their accuracy is strongly correlated with the amount of labeled

data available (Yarowsky and Florian, 2002). Furthermore, obtaining manually labeled corpora with word senses is costly and the task must be repeated for new domains and languages. Ng (1997) estimates that a high accuracy domain independent system for WSD would probably need a corpus of about 3.2 million sense tagged words. At a throughput of one word per minute (Edmonds, 2000), this would require about 27 person-years of human annotation effort. Supervised methods are further restricted by their reliance on a predefined list of senses, which is likely to be inappropriate to a specific domain and task, leading to decreased performance, and necessitating the compilation of a new sense list, and the consequent relabeling of the training data.

Due to these fundamental shortcomings of supervised methods, considerable effort has been devoted to the development of unsupervised methods for WSD, which hold promise for large-scale disambiguation that is unrestricted in terms of language and domain. Unsupervised methods do not require labeled training data, and are therefore much less costly to use, and easier to transfer between domains. The unsupervised framework has the further advantage of allowing more room for the use of linguistic knowledge and resources, and for the application of linguistic theories. In cases where a suitable lexical resource is unavailable (e.g., a new domain with specific terminology that is not contained in standard dictionaries), unsupervised methods can induce the senses directly from the data, thus ensuring that they suit the task at hand. Because of these advantages, there have been many unsupervised WSD methods and approaches proposed in the literature (see Chapter2 for an overview). Despite all this, unsupervised methods have been largely unsuccessful in providing an effective solution to the problem. Their performance falls far below that of supervised methods and is not sufficient to make them useful for real-world applications. For example, in the Semeval English lexical sample task (Pradhan et al., 2007), the best performing unsupervised method achieved an F-Score of 53.8%, compared to 88.7% for the best supervised system. Supervised methods, therefore, are still the solution of choice, despite being unsatisfactory in term of cost and manual labor.

In this thesis we focus on the performance gap separating unsupervised WSD methods from supervised ones. We explore the causes for this gap, and provide solutions to some of the core problems which prevent unsupervised methods from reaching the level of performance and applicability achieved by supervised systems. We hypothesize that the performance gap is largely due to a lack of standardized representation and methodology, which prevents the use of powerful tools for learning, evaluation and combination, such as are common in the field of supervised learning. Another rea-

son for the comparatively weak performance of unsupervised methods is the treatment and evaluation of WSD as a classification task. Under this setting, supervised methods have the advantage of being able to use a large collection of powerful general-purpose classifiers.

These observations guided us in our work to help unsupervised WSD overcome the performance gap. In this thesis, we address and provide solutions for several of the problems we mentioned. For cases where there are existing WSD methods in place, and designing a new WSD system is not feasible or cost-effective, we present a supervised methodology (ensembles) to help improve performance with little need for modification or re-design. In order to more fully exploit the potential of the supervised setting, and for use when suitable existing methods are not readily available, we propose a method for automatically creating sense-labeled training data. This allows the use of powerful supervised classifiers for the task of disambiguation, and greatly reduces the gap between the unsupervised and supervised settings. Finally, we address the issue of treating WSD as a classification task according to a fixed list of senses. We note that for many applications, a fixed dictionary is undesirable, and may not suit the task and domain. Instead, we can take advantage of the freedom offered by the unsupervised setting, which is not restricted by the list of senses used in training. We present a model based on a probabilistic generative formulation which induces the senses directly from the data, thus insuring their relevance to the task and domain at hand. The work presented in this thesis represents a major contribution in providing general-purpose, accurate, WSD methods which are unrestricted in terms of domain, language and application.

1.1 Contributions of the Thesis

In this thesis we address the performance gap separating unsupervised and supervised WSD. As discussed above, this gap represents a major barrier preventing the widespread use and potential benefits of WSD in real-world applications. We take a deep look into the nature of this gap, explore its causes, and present solutions to help bridge it[1]. Our research also provides some important insights regarding the relative strengths and weaknesses of supervised and unsupervised methods in computational linguistics in general. Our individual contributions are detailed below.

[1]Code developed in the process of our work has been made publically available at: `http://homepages.inf.ed.ac.uk/s0570628/code.html`.

A standardized framework for comparison and analysis of unsupervised WSD methods. We present a framework which allows the comparison, evaluation, and detailed analysis of existing unsupervised methods under uniform conditions. This consists of a re-implementation of some of the algorithms and the development of the necessary tools to enable all the component methods to make use of identical input, and produce a similar detailed output format. To our knowledge, this represents a novel setting, and no such detailed comparison has been performed previously. The framework enables an in-depth study of the strengths and weaknesses of individual algorithms, and helps determine important underlying principles which are not specific to a certain algorithm, and can therefore be used in a general setting.

Ensemble combinations. We propose a set of ensemble methods which harness the diversity of existing approaches to unsupervised WSD. These ensembles can be used to improve performance of existing methods with little additional effort. They can also provide a fall-back option in cases where contextual information is not sufficiently informative. We explore several ensembles and show that they can outperform state-of-the-art individual methods. Our work on ensemble methods also represents an important contribution to unsupervised learning in general, since it demonstrates the benefits that can be gained from employing simple methods drawn from supervised methodology in an unsupervised framework.

Unsupervised creation of sense-labeled data. Combination of existing methods provides a solution when a variety of such methods are available, and a quick and easy way of improving results is needed. However, designing a complete WSD method from the ground up offers the possibility of greater performance benefits. Therefore, we develop an unsupervised methodology for the automatic creation of sense-labeled training data. The system makes use of distributional and semantic similarity metrics, and the data it creates can be used to train any standard supervised classifier. We show that classifiers trained on our automatically-created data can surpass the performance achieved using previous methods for data-creation and outperform state-of-the-art unsupervised methods. Our method allows improvements to supervised methods to be easily transferred to the unsupervised setting, since it employs a completely supervised methodology for the actual learning and disambiguation. It therefore represents a significant step in bridging the performance gap between unsupervised and supervised methods. Our experiments also demonstrate the effectiveness of the unsupervised data

creation methodology, and advocate the use of a similar approach for other tasks where manually-labeled data is commonly used.

A Bayesian model for sense induction. The use of a fixed list of senses by all supervised and most unsupervised WSD methods is a serious obstacle to applied WSD, since the predefined sense distinctions are often unsuitable or irrelevant to the task at hand. While supervised methods are naturally constrained to the sense labels used in the training data, unsupervised methods need not restrict themselves in this way. Instead, unsupervised methods can induce the relevant senses directly from the data at hand. We describe a sense induction system which represents an important contribution in this area of unsupervised WSD. We introduce a novel perspective on the sense induction task, which places it in a Bayesian generative context, as apposed to the common approach which treats it as a standard clustering problem. We develop a probabilistic model for the task which provides a principled way of taking into account a wide range of relevant contextual features, and perform an in-depth analysis of the model and its components. Our sense-induction method surpasses state-of-the-art performance on the task. The underlying model is not specific to sense-induction, and can therefore be employed for other tasks where several types of informative features are available.

To summarize, our work explores the nature of the performance gap separating unsupervised and supervised WSD. We address many of the fundamental issues contributing to this gap, and present our solutions to these problems. First, we address the lack of standardization of existing unsupervised WSD systems, and demonstrate how to leverage the diversity with the help of ensemble methods. As our next step, we present a WSD method based on the unsupervised creation of sense-labeled training data. Our system retains the freedom from manual annotation, while avoiding the choice of representation and approach which is a problematic and contentious issue in unsupervised WSD, by handling the disambiguation stage in the supervised setting. Finally, we address the restrictions imposed by a predefined sense inventory by proposing a probabilistic generative model for unsupervised sense induction. This allows unsupervised WSD to be easily integrated into natural-language applications, and tailored to a specific task and domain, without the need to define a new purpose-built sense inventory and corresponding training dataset. All our methods outperform current state-of-the-art unsupervised performance on their respective tasks. While still falling short of state-of-the-art *supervised* methods in cases where manually labeled

training data can be used, they non-the-less represent a significant step in reducing the gap in WSD and enabling large-scale use of unsupervised WSD in real-world applications, where such data is usually unavailable.

1.2 Thesis Structure

In Chapter 2 we familiarize the reader with the field of unsupervised WSD. First, we introduce relevant terminology and present some resources commonly used in the field and employed in this thesis. These include standard sense inventories, large scale corpora, and evaluation resources. We conclude the chapter with a high-level overview of previous work, with references to the appropriate sections in following chapters, each containing a more detailed overview of work directly relevant to that chapter.

In Chapter 3 we look into the reasons for the performance gap between supervised and unsupervised methods for WSD. We design a framework for standardized analysis and evaluation of unsupervised WSD systems. We select four methods representing different approaches to unsupervised disambiguation: a simple context-overlap approach, two methods employing different graph-based representations, and an algorithm which uses vector-based distributional and semantic similarity measures for determining the predominant sense of an ambiguous word in a corpus. We compare them in detail, examining their strengths and weaknesses and the differences between them. Our experiments reveal that there is only a small overlap between the methods in terms of correctly disambiguated words. We therefore, in the second part of the chapter, present a set of ensembles, an idea borrowed from supervised methodology, in order to leverage this complimentary nature. Our ensemble methods outperform the individual component methods, and produce state-of-the-art results on standard evaluation datasets.

In Chapter 4 we develop an automatic method for unsupervised creation of sense-labeled training data. This approach provides the means of circumventing many of the problems leading to the performance gap between unsupervised and supervised WSD, such as lack of a standard representation or powerful, well-studied tools for classification. Our method, as opposed to previous methods of automatic data annotation, makes minimal use of linguistic resources, and is thus applicable to a wider range of domains and languages. In order to assess the usability of our automatically created data, we use it to train a selection of classifiers representing different machine-learning paradigms, and compare them on two standard evaluation datasets. We also compare

to previously proposed methods for automatically creating sense-labeled data. The results achieved using our method are significantly better than those achieved when using previous ones, and approach the performance of training on manually annotated data.

In Chapter 5 we direct our efforts towards another fundamental drawback of supervised methods, namely their reliance on a predefined list of senses. We address the task of inducing senses from the data, independently of any dictionary, by developing a system based on a generative probabilistic model, Latent Dirichlet Allocation (LDA). We adapt the model (originally designed for modelling text generation) to the sense-induction problem, and extend it to take into account multiple sources of information relevant to the task. The new model represents a general extension of LDA, and can be used for a variety of tasks where multiple information sources are available. We explore the use of our model for sense induction, looking into several relevant issues, such as the nature of the learning corpus, the choice of information sources, and the importance of parameter tuning. Comparison on a standard evaluation dataset shows that our sense-induction system outperforms other state-of-the-art methods on this task.

We conclude in Chapter 6, with a summary of the main findings of this work. We also discuss possible applications for the methods presented in the thesis, and directions for further research.

1.3 Published Work

Parts of the work presented in this thesis have been previously published. This applies to Chapter 3, portions of which have been published in ACL-COLING (Brody et al., 2006), and Chapter 4, in COLING (Brody and Lapata, 2008).

Chapter 2

Related Work

I have read your book and much like it.

Moses Hadas

Before proceeding to the main body of the thesis, and presenting our methods and contributions to the problem of unsupervised WSD, it is necessary to familiarize the reader with the some background knowledge about the problem and the setting in which it is addressed. This chapter fulfills this function. We start with a general discussion regarding levels of supervision in machine learning in general, and WSD in particular (Section 2.1). We then introduce relevant terminology from the field of WSD that is used in this thesis (Section 2.2). In Section 2.3, we provide a description of data and evaluation resources commonly used in unsupervised WSD, and of which we make use in our work. Finally, in Section 2.4 we describe related previous work. Since the methods presented in this thesis fall naturally into three individual subsets of the field of unsupervised WSD, we chose to place each piece of work within the relevant context at the beginning of each chapter. However, Section 2.4 provides a general overview of the field, and indicates where each subset we address falls within the field as a whole. It thus links together our individual discussions of related work into a single unit.

2.1 Annotation and Supervision

In many fields of computer science, it is customary to distinguish between three settings for machine learning: (1) supervised, (2) unsupervised and (3) semi-supervised. In the supervised setting, the computer program is provided with labeled training data before it receives the unlabeled test data. The program attempts to extract from the

labeled data information on how to provide the correct labels for the items in the test set. In the unsupervised setting, on the other hand, no labeled data is provided. The program is expected to detect *"natural"* distinctions in the test data which divide it in a significant fashion according to some criteria. The labels the program provides in the output have no meaning besides serving to distinguish between the classes that were detected. The semi-supervised setting is midway between the previous two. In this setting, a small amount of labeled data is provided, along with a large amount of unlabeled data. The quantity of the labeled portion of the data is usually not sufficient to permit reliable learning. A semi-supervised algorithm must make use of both the high accuracy information about the labels provided by the labeled data, and the information about the global structure of the data learned from the large unlabeled dataset. The algorithm combines all this information to try and match the distinctions represented by the provided labels with natural divisions it detects in the data.

2.1.1 Supervision in WSD

In the field of WSD, the same terms are often used, but with slightly different meaning. Since WSD is almost always evaluated with respect to a given sense inventory, there is effectively a small amount of labeled data inherent in the task. Even unsupervised methods for WSD are expected to produce output which is tagged with meaningful senses from the sense inventory. Unsupervised WSD methods using a specific sense inventory are therefore more similar to semi-supervised methods in other fields. The exception to this are sense induction methods where the sense-inventory is not predefined, and the output of the program is not expected to match a gold standard labeling. In this case, the labels themselves have no external meaning, matching only natural divisions in the data with respect to the task. These cases therefore conform to the classic definition of an unsupervised setting.

Since most unsupervised WSD methods *do* have a small amount of highly informative data (the sense inventory and sense definitions therein), many unsupervised methods tend to make heavy use of this data and knowledge of linguistic theory (see, for instance, Section 3.2) when attempting to solve the WSD problem.

Supervised WSD, on the other hand, is much more like the classic supervised setting – an amount of annotated text is provided as input to the algorithm, which is used to learn how to label new unlabeled data. The sense-inventory itself is usually only used to provide the set of valid labels, and then ignored by the algorithm. Supervised

techniques do not need to resort to semantic information to match meanings in the text to those in the sense inventory. Since the labels are provided by the annotated text, a simple vector representation, such as is used in many machine learning classification algorithms, is sufficient. As a result, the data representation is more-or-less identical between the different methods. Data instances are represented by a feature vector extracted from the context of the word, and the precise identity of the feature set is largely independent of the method employed. The main differences between approaches, and the points at which linguistic knowledge comes in, are (1) the choice of machine learning technique to employ to best capture the distinctions in the data, and (2) the choice of which feature space would be most informative for the task. Mooney (1996) discusses in detail the significance of these choices. He also provides a comparison of several machine learning methods applied to WSD. Yarowsky and Florian (2002) compare some more recent efforts to use machine learning techniques for WSD.

To summarize, there are three main settings in the field of WSD. In the *supervised* setting, machine-learning classifiers are trained on examples which are annotated with the correct sense in accordance to predefined list of senses (usually a dictionary). These methods are the most accurate, but require manual annotation which is very expensive in term time and effort. In the *unsupervised* WSD setting, the methods are required to label the instances with the correct sense from the dictionary. No labeled examples are given, but the algorithms can make use of the information contained in the dictionary, and other lexical resources, as well as unlabeled corpora. The third setting is *sense induction*, which is a special case of unsupervised WSD, where no dictionary is given (and, of course, no labeled examples). The methods is expected to induce the relevant sense distinctions from the data itself, and then label the instances accordingly. In this thesis we focus on the latter two settings: standard unsupervised WSD and (unsupervised) sense induction. Supervised methods provide an upper bound on expected performance.

2.2 Terminology

Sense Inventory A resource containing a list of possible senses and their definition, for each word of interest. Sense inventories are often standard dictionaries, but can also be of other forms (e.g., thesauri, which list words with similar meaning for each entry). Sense inventories differ with regard to the amount and the nature of the information they provide.

Word Sense Disambiguation The task of assigning one of several possible sense-labels to a word. The list of possible senses for each word is fixed, and contained in a pre-specified sense inventory, which also provides the definitions of the senses and possibly further information, such as examples or related words. The term *disambiguation* is sometimes used in a wider sense to include sense *discrimination*, where the senses are not pre-defined (see below). In this thesis, we will distinguish between the two.

Supervised WSD Disambiguation with the aid of labeled training examples. The sense labels conform to a predefined sense inventory.

Unsupervised WSD Disambiguation according to a predefined sense inventory, but without labeled training examples. Information from lexical resources (primarily the dictionary serving as the sense inventory) and additional corpora of unlabeled text are used.

Sense Induction / Sense Discrimination The task of separating the different occurrences of a given word into two or more groups (or *clusters*) representing different meanings. In this task, as opposed to word sense disambiguation, the number and identity of the possible senses is not pre-defined, and must be inferred by the algorithm. No dictionary or other description of the senses is provided.

Knowledge-Rich Methods WSD methods which make use of a pre-defined and fixed sense-inventory, and possibly other linguistic knowledge and resources. Most unsupervised WSD methods fall into this category, and the term is used to emphasize the fact that they are only *unsupervised* in not using labeled training data. However, they make use of other knowledge sources.

Knowledge-Lean Methods Typically sense-discrimination methods that do not make use of a sense inventory or other lexical resources. The term is used to distinguish these methods from the more common, knowledge-rich, unsupervised WSD methods.

Word Tokens vs. Word Types The distinction between a word *type* and a word *token* was first made by Peirce (1933). The type of a word is the abstract notion of that word, whereas a token is the representation of that word as used in a particular point in the text. The author further distinguishes between the a *token* and an *instance*, where the

token is the physical representation of the word type, and the instance represents the individual point at which it occurred. For example, in the sentence *"The quick brown fox jumped over the lazy dog"* there are two instances of the token *the* which embodies the type (abstract word) *the*. It is common practice (though slightly inaccurate), to use *token* to signify an *instance*. Our example, therefore, is said to contain two *tokens* of the single type *the*. Since this distinction is more relevant to our work, we will follow this practice throughout the thesis, and refer to instances as *tokens*, and to the abstract notion of the word as its *type*.

Target Word The word that is the current focus of the disambiguation (or discrimination) algorithm, as opposed to other words in the context or in the lexicon. For instance, in the sentence *"The quick brown fox jumped over the lazy dog"*, we may wish to determine whether the word *fox* refers to an animal or a person of shifty nature. In this case, our target word is *fox*, and the rest of the words in the sentence are merely helpful contextual clues for disambiguation of the target.

Coarse-Grained vs. Fine-Grained WSD Sense-inventories (see above) have differing opinions as to what constitutes a sufficient distinction between two senses, or conversely, when two shades of meaning are similar enough to fall into the same *sense*. Some sense inventories (e.g., Oxford Dictionary of English) employ a hierarchical structure, where the top level corresponds to coarse-grained distinctions, and lower levels distinguish between increasingly fine shades of meaning. It is widely recognized (see Edmonds and Kilgarriff 2002; Navigli 2006; Snow et al. 2007) that differing levels of granularity are suitable for different tasks. For many applications (e.g., information retrieval) coarsely defined senses may be more useful (see Snow et al. 2007 for discussion). For example, the word *sense* has five senses in WordNet, of which two were grouped together by the annotators in the Senseval 2 workshop (see Section 2.3.3.2), to form the following four coarse-grain senses.

1. a. A general conscious **awareness**.
   (e.g., *a sense of security*)

   b. The faculty through which the external world is apprehended.
   (e.g., *a sense of smell*)

2. The **meaning** of a word.
   (e.g., *The dictionary gave several senses for the word*)

3. Sound practical **judgment**.
   (e.g., *I can't see the sense in doing it now*)

4. A natural appreciation or **ability**.
   (e.g., *keen musical sense*).

Senses 1a and 1b represent different aspects of the same general meaning – that of feeling or perception.

Co-occurrence Two words are said to co-occur if they both appear within a specified distance (window) of one another. Distance can be measured in terms of words, sentences, paragraphs, etc. For instance, in the sentence *"The quick brown fox jumped over the lazy dog"*, the words *quick* and *fox* co-occur within a 3-word window, but *fox* and *lazy* do not, since they are more than three words apart. They do, however, co-occur within the same sentence. The *Distributional Hypothesis* (Harris, 1985) posits that words which tend to co-occur with a target word provide a strong indication of its meaning. Therefore, features based on co-occurrence counts are common in the field of WSD.

Distributional Similarity Based on the *Distributional Hypothesis* (see above), words which have similar patterns of co-occurrence (tend to co-occur with the same words) are expected to have similar meaning. Distributional similarity metrics are ways of quantifying this type of similarity. They assume a vector representation (see Section 3.2.3) based on some form of co-occurrence information, and provide the means of quantifying the similarity (or, conversely, the distance) between the two words, by comparing their vector representations.

Semantic/Dictionary-Based Similarity A method for measuring similarity between words (or between word-senses) using the information provided in a lexical resource, such as a dictionary. The WordNet Similarity package (Pedersen et al., 2004), is one of the most popular collections of such methods, providing implementations of several semantic similarity metrics for WordNet (see Section 2.3.1).

2.3 Resources

2.3.1 Sense Inventories

In order to adequately define the task of disambiguation, an agreed-upon sense inventory must be established. Such an inventory provides the list of ambiguous words, and defines the possible senses for each word.

Traditionally these sense inventories have been well-known dictionaries, such as the Oxford Dictionary of English and Longman's Dictionary of Contemporary English. Several WSD methods (e.g., Yarowsky 1992; Mohammad and Hirst 2006) have also made use of Thesauri, such as Roget's Thesaurus and the Macquire Thesaurus. Dictionaries define the senses using glosses in natural language format, whereas thesauri define senses in terms of synonymous words. The latter format is easier to process computationally. Unfortunately, traditional thesauri were not designed as sense inventories to assist in language comprehension (this task was left to dictionaries), but rather to provide assistance for humans when writing. This fact affected the way the thesauri were constructed, which senses were considered and included, and what information was provided in the entries.

With the advances of computational linguistics, many well established sense inventories have been converted to machine readable form. There have also been new resources designed from the start with the aim of providing data for computational processing of language. One of the most widely-used resources in the NLP community is WordNet (Fellbaum, 1998). This resource is essentially a dictionary and thesaurus combined, represented in a graph-like structure. English nouns, verbs, adjectives and adverbs are organized into synonym sets (graph nodes), each representing one underlying lexical concept. These *synsets* are linked together by labeled edges representing different linguistic relations. These relations are primarily hypernymy/hyponymy (superordinate/subordinate), antonymy, entailment, and meronymy/holonymy. There are also links representing derivationally related forms, attributes and "see-also" relations. The noun *brother* has the following entries in WordNet:

- Sense 1: brother, blood brother (a male with the same parents as someone else) "my brother still lives with our parents"
- Sense 2: brother (a male person who is a fellow member (of a fraternity or religion or other group)) "none of his brothers would betray him"

- Sense 3: buddy, brother, chum, crony, pal, sidekick (a close friend who accompanies his buddies in their activities)
- Sense 4: brother, comrade (used as a term of address for those male persons engaged in the same movement) "Greetings, comrade!"
- Sense 5: Brother ((Roman Catholic Church) a title given to a monk and used as form of address) "a Benedictine Brother"

Each entry contains a list of synonymous words and a gloss (in parenthesis) describing the meaning shared by the members of the synset. Many synsets also give one of more examples of usage (in quotation marks). Each synset also lists the relations in which it takes part. Nouns and verbs are organized into hierarchies based on the hypernymy/hyponymy relation between synsets. An example of a portion of the noun hierarchy is shown in Figure 2.1. The synsets representing the first sense of *brother* are linked through the antonymy relation to the synset of (the first sense of) *sister* . The synset for *bone* is linked to *arm* and *leg* through the meronymy (part-of) relation, and to *organic substance* through the hyponymy (is-a) relation.

Adjectives are arranged in clusters containing head synsets (an antonymous pair, or occasionally a triplet) and satellite synsets, representing concepts that are similar in meaning to the concept represented by a head synset. Adverbs are often derived from adjectives, and sometimes have antonyms. Therefore the synset for an adverb usually contains a lexical pointer to the adjective from which it is derived.

The first publicly-available version of WordNet was 1.5 (released in 1995). This was created by a team of lexicographers, basing themselves primarily on words and senses found in the SemCor corpus (see Section 2.3.3.1). Over the years, there have been several versions of WordNet, as more lexical items were added and changes and revisions were made. The latest version (WordNet 3.0) contains a total of 155,287 unique strings, divided into 117,659 synsets. These are composed of four part-of-speech portions. The noun portion contains 117,798 words and 82,115 synsets. The verb portion has 11,529 words and 13,767 synsets. The adjective portion has 21,479 words and 18,156 synsets, and the adverb portion 4,481 words and 3,621 synsets.

WordNet has been widely used in WSD research, both as a sense inventory (see discussion in Kilgarriff and Palmer 2000), and as a knowledge resource in WSD algorithms (for examples, see Section 3.2).

WordNet enjoys widespread popularity, and many tools have been designed which make use of its data. One such resource is the WordNet Similarity package (Pedersen

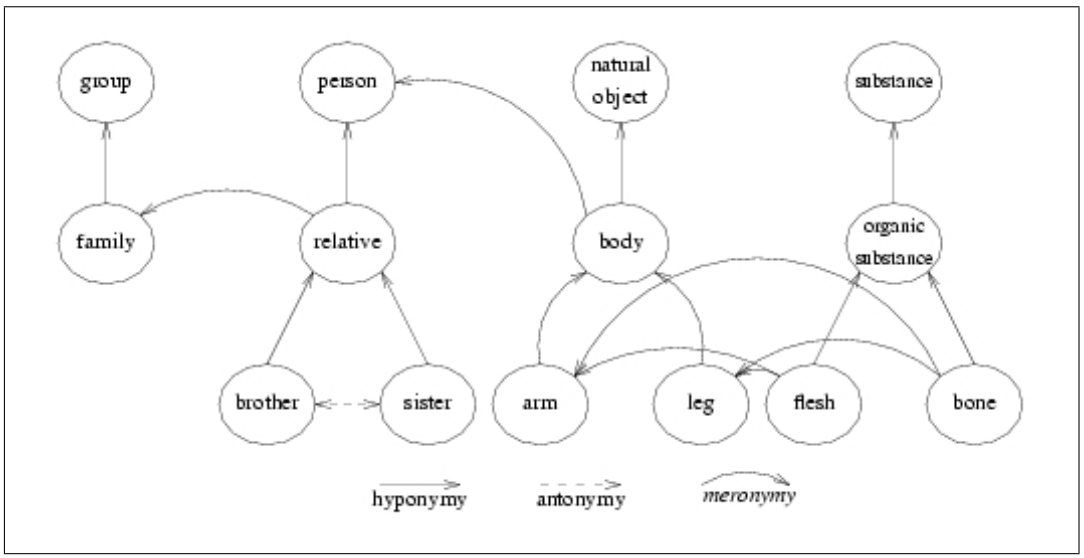

Figure 2.1: An example of three semantic relations (hyponymy, antonymy and meronymy) in a portion of the WordNet noun hierarchy. From Miller et al. (1990).

et al., 2004). This resource contains implementations of different methods which have been proposed for automatically measuring the semantic similarity between synsets in WordNet. These include methods which rely on the WordNet graph structure, corpus statistics, the WordNet glosses and usage examples, and on combinations thereof. For a good overview of the different methods, see Pedersen et al. (2004), and for an in-depth comparison see Budanitsky and Hirst (2004).

Despite the widespread use of WordNet, and the enormous amount of information contained therein, there are several problems with its use for WSD. One problem is the absence of a similar resource in other languages. Though there are attempts to create WordNets in other languages (e.g., Vossen 1998), the effort involved is enormous, and for many less widely-spoken languages there is little hope of a similar resource in the near future. Another criticism is that the division of senses in WordNet is often extremely fine-grained (Edmonds and Kilgarriff, 2002), and therefore can be more of a hindrance than an assistance in many real-world applications. Often the disadvantage of the increase in data sparseness and the difficulty in automatically detecting fine distinctions outweigh the small benefit these distinctions provide to the application (see Snow et al. 2007).

There have been several attempts (e.g., Agirre and de Lacalle 2003; Navigli 2006) to provide a coarser-grained division of WordNet synsets, by unifying groups of synsets which are only marginally distinct. Ideally, such groupings should be done when creating the sense inventory. When this was not done, producing such a grouping manually

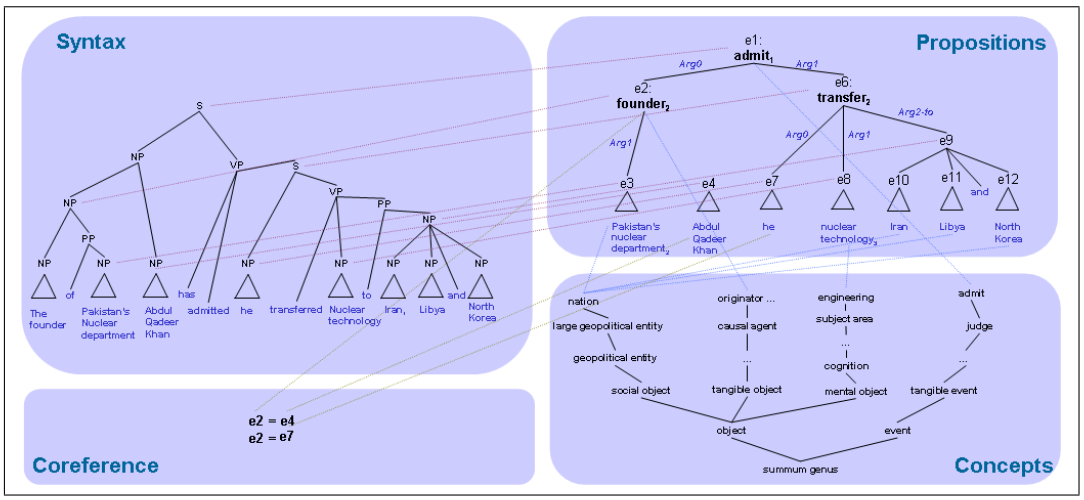

Figure 2.2: An example of an annotated sentence and related structure from OntoNotes. The sentence structure is represented with a syntactic tree (upper-left). The argument structure of the verbs in the sentence is indicated (upper-right), co-referring entities are resolved (lower-left), and the portions of the concept hierarchy relevant to the sentence are shown (lower-right). Corresponding elements between these structures are linked.

is usually infeasible, considering the size of a wide coverage sense-inventory. Instead, it is necessary to rely on automatic methods, at the cost of reduced accuracy.

The OntoNotes project[1] (Hovy et al., 2006) is a collaboration between several universities and companies. It aims to provide a publicly available annotated corpus comprised of various genres of text (news, conversational telephone speech, weblogs, use net, broadcast, talk shows) in three languages (English, Chinese, and Arabic). The text is annotated with structural information (syntax and predicate argument structure) and shallow semantic information (i.e., word senses linked to an ontology and co-reference). These layers of annotation, all making use of a common ontology and indexing system, provide a level of semantic representation far beyond the entity and relation types annotation presently in use in many tasks. An example of an annotated sentence, along with the relevant structure, is shown in Figure 2.2. The creators put a strong emphasis on the quality of annotation, realizing the importance this has when used to train machine learning algorithms. Therefore, they aim to ensure that every layer of annotation has at least 90% inter-annotator agreement. Pilot studies they performed have shown that predicate structure, word sense, ontology linking, and coreference can all be annotated rapidly and with better than 90% consistency. The creators

[1] `http://www.bbn.com/ontonotes`

of OntoNotes hope it will fundamentally change the field of natural language processing, similarly to the Penn TreeBank and WordNet, and enable applications to break the current accuracy barriers in transcription, translation and question answering.

2.3.2 Corpora

Large-scale machine readable corpora provide an important resource in unsupervised WSD. Since unsupervised algorithms are not provided with labeled training examples, they must learn as much as possible from other sources. Large-scale corpora can provide important information about the characteristics of the data on which the algorithms will be evaluated, as well as statistical properties of the words (e.g., frequency, common usage patterns). In this section we present the corpora we used in our work.

2.3.2.1 British National Corpus (BNC)

The British National Corpus (Clear, 1993) is a 100 million word collection composed of 4049 samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of British English from the later part of the 20th century, both spoken and written. The written part (90%) contains a wide variety of text from different domains. It includes, among other sources, newspaper articles, journals, academic literature and fiction, letters, and school and university essays. The spoken part (10%) consists of transcription of informal conversation (selected in a demographically balanced way to represent different ages, regions and social classes), and spoken language from a range of contexts, from formal meetings to radio shows. The BNC is a widely used resource for natural language processing.

2.3.2.2 Wall Street Journal (WSJ)

The Wall Street Journal[2], is a business publication providing news with a strong financial and business aspect. As a publicly available corpus, the Wall Street Journal comes in two parts. The first, containing approximately 30 million words and composed of articles from 1987-1989, has been widely used in NLP. It is the base of the manually annotated DSO (Ng and Lee, 1997) , Penn Treebank (Marcus et al., 1993), and PropBank (Palmer et al., 2005) corpora. The second part, containing articles from the years 1994-1996, approximately 40 million words, is available as part of the North American News Text Corpus (Graff, 1995).

[2] `http://online.wsj.com`

2.3.3 Evaluation Datasets

An important part of the study of word sense disambiguation is the availability of standard evaluation datasets and metrics. These allow reliable quantification of performance and accurate comparison between methods under identical conditions. It is therefore important to present the evaluation resources we used in our work.

2.3.3.1 SemCor

The SemCor corpus[3] (Landes et al., 1998), created by the Princeton University, is a subset of the English Brown corpus (Kucera and Francis, 1967). It is composed of 352 texts, and contains approximately 700,000 running words. For 186 texts (more than 200,000 content words), all open class words (nouns, verbs, adjectives, and adverbs) are annotated with part-of-speech, lemma and sense information, while in the remaining 166 texts only verbs are annotated with lemmas and senses. In total, the "all-words" component of SemCor has 359,732 tokens among which 192,639 are semantically annotated, while the "only-verbs" component has 316,814 tokens, among which 41,497 verb occurrences are semantically annotated. WordNet version 1.6 was used as the sense inventory, but the annotation has been automatically mapped to all later versions of WordNet.

The corpus was created to guide the WordNet annotators with regard to the possible senses in context, discover words and senses missing from WordNet, and provide examples of the senses in context. The order of senses in WordNet is according to their frequency in SemCor. Senses that do not appear in the corpus are ordered arbitrarily, after those which are attested (McCarthy et al., 2004). Information about the sense frequencies of a word, and particularly the identity of the first sense in WordNet (i.e., the most frequent sense in SemCor), has been used extensively in both supervised (e.g., Hoste et al. 2001) and unsupervised (e.g., Galley and McKeown 2003) WSD algorithms, as a fallback in cases where local contextual information is insufficient.

SemCor is one of the largest corpora which is sense-annotated for all words, and along with its matching sense-inventory, WordNet, provides a standard evaluation resource for WSD algorithms.

[3] SemCor is publically available at `http://www.cs.unt.edu/rada/downloads.html`.

2.3.3.2 Senseval and Semeval

Much of the progress in WSD is due to workshops and evaluation exercises organized by Senseval, an international organization devoted to the evaluation of word sense disambiguation systems. Its purpose is stated in the *Constitution of Senseval*[4]:

- To organise activities for the evaluation of word sense disambiguation programs.
- To promote interest in the Lexicon and Word Sense Disambiguation (WSD).
- To provide members of the ACL and ACL-SIGLEX having a special interest in WSD with a means of exchanging news of recent research developments and other matters of interest.
- To sponsor meetings and workshops on WSD and related themes that appear to be timely and worthwhile.

The constitution of Senseval also states that along with understanding of the importance of WSD in application of language technology, the organization's underlying goal is to the further the understanding of lexical semantics and polysemy. In order to study and evaluate WSD, Senseval has primarily focused on stand-alone WSD, despite acknowledging that WSD, in many applications, is an inseparable part of a complex system. The SENSEVAL organization was started in 1997, following a workshop, *"Tagging with Lexical Semantics: Why, What, and How?"*, held at the conference on Applied Natural Language Processing.

Senseval-1[5] (Kilgarriff and Palmer, 2000) was held in the summer of 1998, culminating in a workshop at Herstmonceux Castle, England. Following the success of the first workshop, Senseval-2[6] (Preiss and Yarowsky, 2001) was held in 2001, in conjunction with ACL in Toulouse. Senseval-2 included tasks for twelve languages, including Chinese, Dutch, Estonian and Korean. Senseval-3[7] (Mihalcea and Edmonds, 2004) took place in 2004, followed by a workshop held later that year in Barcelona, in conjunction with ACL. Senseval-3 included 14 different tasks for core word sense disambiguation, as well as identification of semantic roles, multilingual annotations, logic forms and subcategorization acquisition. Semeval-1/Senseval-4[8] (Agirre et al., 2007) took place in Prague in June 2007, in conjunction with ACL. There were 19 tasks,

[4] http://www.senseval.org/overview.html
[5] http://www.itri.brighton.ac.uk/events/senseval/ARCHIVE/index.html
[6] http://www.sle.sharp.co.uk/senseval2
[7] http://www.senseval.org/senseval3
[8] http://nlp.cs.swarthmore.edu/semeval

including one on cross-language information retrieval and one on the evaluation of sense-discrimination, as well as the standard stand-alone all-words and lexical-sample evaluation tasks. All Senseval workshops have made their datasets available in the public domain.

2.3.3.3 All-Words vs. Lexical Sample

WSD algorithms are commonly evaluated on two tasks: all-words and lexical-sample. In the all-words setting, the systems being evaluated must disambiguate all the (content) words in a given piece of text, according to a sense-inventory containing a list of possible senses for each word. In the lexical sample setting, a list of words is selected by the task designers, and the systems are evaluated according to their performance on (only) these words. While the all-words task is a more representative of overall WSD performance, it has several problems. From a pragmatic perspective, creating an all-words sense-labeled corpus is very labour-intensive, and even large quantities of text contain only few instances of low-to-medium frequency words, making accurate evaluation difficult. Also, it is not clear that all-words is the task for which a WSD system would be applied in a real-world application. For purposes of information extraction and question answering, for instance, it might be more practical to focus on a few words whose variation in meaning strongly influences the results.

2.4 Related Work Overview

A detailed overview of the entire field of WSD is unfeasible in this context. We chose instead, at the beginning of each chapter, to provide detailed descriptions of previous work that is directly related to that presented in the chapter. This provides a natural division of related work in the field into three main categories. In Chapter 3 we give an overview of classic unsupervised WSD methods, which make use of a standard sense inventory and lexical knowledge in a variety of approaches. In Chapter 4, we address methods for unsupervised creation of labeled data and semi-supervised bootstrapping techniques which lie on the boundary between supervised and unsupervised methods. Finally, in Chapter 5, we describe methods for sense-induction, which is a completely unsupervised setting and does not make use of a predefined sense inventory or lexical resources. Examples for each setting are provided in each case. Together, the related-work sections of each chapter provide the background for the issues which are the

focus of this thesis – the nature of the gap between supervised and unsupervised WSD techniques, and the steps that can be taken to diminish it.

2.5 Summary

In this chapter, we provided some necessary background regarding the problem of word sense disambiguation, and its research. We introduced relevant terminology and described data and evaluation resources from the field which we use in this thesis. We concluded with an overview of related previous work, which links together the specific areas of the field discussed in later chapters, and places them in a global context.

Chapter 3

Ensemble Methods for Unsupervised WSD

3.1 Introduction

As stated in the introduction to this thesis, unsupervised methods hold the key to enable wide-scale WSD for real world applications, since they are not restricted by the need for sense-labeled data. However, unsupervised methods fall far behind supervised ones in accuracy. Our goal in this thesis is to make use of methods and ideas from the supervised learning framework to close the gap and help bring unsupervised WSD closer to the accuracy achieved by supervised methods. In order to address this issue, we must ask ourselves: what is the source of the accuracy gap? Is the problem in the algorithms, in the choice of inventory, the data, or all of the above? In this chapter, as a first step, we examine the issue of formulation and algorithms (leaving the issues of data and sense-inventory for Chapters 4 and 5, respectively). We compare and contrast existing methods for unsupervised WSD in order to answer the following questions: (1) Do existing methods yield similar results? (2) Are they complementary and, if so, how can they be combined? (3) What are the key elements in successful methods?

Methods for unsupervised WSD vary greatly in approach (e.g., type vs. token approaches, see next section), formulation of the problem (e.g., graphical vs. vector space representation), and type of knowledge used (e.g., semantic relatedness vs. distributional similarity). In Section 3.2 we provide a detailed discussion of these issues along with examples. These differences result in a lack of standardization and compatibility between different WSD algorithms which makes comparison and integration between different algorithms very difficult. On the other hand, the variety of

approaches presents a wide spectrum of unsupervised WSD algorithms, with possible complementary aspects. Our comparison shows that the algorithms we examine yield sufficiently diverse outputs, thus motivating the use of combination methods for improving WSD performance. We present a method for leveraging this diversity by combining the outputs as an ensemble in an unsupervised fashion. We thus take advantage of currently existing methods and resources without resorting to difficult modifications of the algorithms themselves. Combination approaches are a common tool in the supervised framework, and have been studied previously for supervised WSD (Florian et al., 2002). However, their use in an unsupervised setting is, to our knowledge, novel. We examine several existing and new unsupervised combination methods and demonstrate that our combined systems consistently outperform the state-of-the-art (e.g., McCarthy et al. 2004). Importantly, our WSD algorithms and combination methods are completely unsupervised, and do not make use of training material in any way, nor do they use the first sense information available in WordNet.

This chapter consists of two main parts. In the first part, we provide an overview of existing approaches to unsupervised WSD, and describe in detail the main algorithms from each approach. Next, we present a detailed comparison of the performance of a representative group of methods on SemCor (Miller et al., 1993). In the second part, we motivate the use of combination methods to harness the diversity of existing approaches in order to improve unsupervised WSD accuracy. We evaluate our unsupervised ensemble methods, and show that they outperform state-of-the-art individual methods. We conclude with a discussion of our results and findings, and directions for future research.

3.1.1 Types vs. Tokens

An important issue in the task of WSD is that of type- versus token-based approaches. Token-based approaches consider each occurrence of an ambiguous word independently, and use its immediate context for disambiguation. However, it has been observed that, in many cases, texts tend to follow the 'one-sense-per-discourse' rule (Gale et al., 1992b). Human writers tend to restrict themselves to a specific sense of a word they use throughout a piece of text, since switching senses in the middle of a discourse creates confusion for the reader and hinders comprehension. This tendency can be exploited to help resolve ambiguity, and provides the foundation for the type-based approach. If a certain instance of a word has highly ambiguous context, other instances

of the same word (the same *type*) in the document can be used to infer the most probable sense, since they are likely to share it. Therefore, if we know that most of the other occurrences of the word in the document have sense *s*, we can assign that sense to instances were we are less certain. Type-based methods go even further, and use this as the main strategy for disambiguation. They consider all occurrences of a word (all tokens of a single type) in the document as one unit, and determine a single sense for it. The 'one-sense-per-discourse' approach assumes a single most probable sense in a given discourse or document. The 'predominant-sense' approach (McCarthy et al., 2004), takes this assumption a step further. It assumes a single, most frequent, sense for each ambiguous word in a given corpus or domain. Many WSD algorithms use the first sense from WordNet (which is the most frequent sense in the SemCor corpus) as a fallback option when the method encounters an instance where the correct sense cannot be reliably estimated (e.g., Galley and McKeown 2003, Hoste et al. 2001). This fallback method has been shown to be very effective and in many cases outperforms unsupervised WSD methods (McCarthy et al., 2004). However, the predominant sense of a word can change from domain to domain. For example, the predominant sense of the word *bill* depends on whether we are in the legal domain (where *bill* is likely to be a statute in draft) or financial domain (where the *bank-note* sense is prominent), and the most frequent sense of *strike* may differ when we move from a sports domain (a term from baseball) to an employment/labor domain (the *refusal-to-work* sense). Estimating the predominant sense in a given domain has been shown to be a difficult but worthwhile task with regard to WSD (Koeling et al., 2005).

3.2 Unsupervised Approaches

This section provides a brief survey of a variety of approaches to unsupervised WSD, with the intent of familiarizing the reader with the diversity of existing methods addressing the task. We chose to divide the approaches by the way they represent the WSD problem. We describe three main categories: (1) Direct Context, (2) Graph-based, and (3) Vector-based. The Direct Context approach is the most simple, and does not make use of underlying structure. It works purely at the word level, and makes only minimal assumptions about the connection between context and meaning. Graph-based approaches, on the other hand, make strong structural assumptions. They try to model the underlying semantic connection between words in the context, and make heavy used of linguistic knowledge for this purpose. The methods then lever-

age this structure to disambiguate and infer meaning. Finally, vector-based methods approach the problem from a statistical, mathematical view. They posit that mathematical similarity/distance metrics in an appropriate vector space can by used to infer similarity in meaning, and help to resolve ambiguous cases. While this approach also makes heavy use of the connection between context and meaning, it does so implicitly, in the design of the vector-space, and makes fewer assumptions as to semantic structure. We present a detailed description of a few disambiguation methods demonstrating the implementation of each approach.

3.2.1 Direct-Context Approach

3.2.1.1 Extended Gloss Overlap

Gloss Overlap was originally introduced by Lesk (1986) for performing token-based WSD. It is one of the earliest and most basic methods proposed for unsupervised WSD, and the foundation of many WSD algorithms. The method assigns a sense to a target word by comparing the dictionary definitions of each of its senses with the words in the surrounding context. The sense whose definition has the highest overlap (i.e., words in common) with the context is assumed to be the correct one.

Banerjee and Pedersen (2003) present the concept of "extended glosses". They augment the dictionary definition (gloss) of each sense with the glosses of related words and senses. These *extended glosses* increase the information available in estimating the amount of overlap. The original Extended Gloss Overlap measure presented in the article compares the extended glosses of two senses. The authors present a WSD algorithm which uses this measure for disambiguation of a word in context by calculating the extended-gloss overlap between each sense of the target word and the extended glosses of all the senses of all words in the context. The sense of the target which has the highest overall overlap is chosen as the correct one.

Due to the large amount of relations considered by the algorithm, it is computationally complex, and expensive in terms of running time and computational resources. A much simpler algorithm, based on the same idea, but closer to the original method proposed by Lesk (1986), compares the extended gloss of each sense of the target word directly to the surrounding context, rather than using the glosses of the individual context words. In this work, we focus on our simpler version of the Extended Gloss Overlap algorithm.

As an example, consider the word *arrow* for which WordNet contains two senses:

1. a mark to indicate a direction or relation.

2. a projectile with a straight thin shaft and an arrowhead on one end and stabilizing vanes on the other; intended to be shot from a bow.

In the sentence "*He shot the arrow, scoring ten points.*" the original overlap method would detect the overlapping word *shot* occurring in both the sentence and the gloss of the second sense, allowing the algorithm to correctly label the word with that sense. However, in the sentence "*He fired the arrow at the target.*", no overlap (disregarding function words) exists between the sentence and either of the glosses. However, if we consider the extended gloss of the second sense, containing the gloss of its hypernym *projectile*: *"A weapon that is forcibly thrown or projected at a target but is not self-propelled"*, the overlap with the word *target* in the above sentence gives us the correct sense.

The range of relationships used to extend the glosses is a parameter, and can be chosen from any combination of WordNet relations. In their implementation, Banerjee and Pedersen (2003) make use of all first-order WordNet relations, considering hypernyms, hyponyms, holonyms, meronyms and words related through the 'attribute', 'see-also', 'similar-to', 'entailed-by' and 'cause' relation indicators. For every sense s_k of the target word, the following score is calculated:

$$SenseScore(s_k) = \sum_{Rel \in Relations} Overlap(context, Rel(s_k)) \tag{3.1}$$

where $context$ is a simple (space separated) concatenation of all words in a context window of length $\pm n$ around the target word w_0 (i.e., all w_i for $-n \leq i \leq n, i \neq 0$), and $Rel(s_k)$ is the gloss (or glosses) of the synset(s) related to s_k through relation-type Rel. The overlap scoring mechanism is also parametrized and can be adjusted to normalize the length of the glosses, to exclude examples from the glosses, or to ignore function words.

3.2.2 Graph-Based Methods

Graph-based methods share some common elements. They all represent the context as a graph where the nodes are word-senses and the edges represent semantic connections between them. They then use different characteristics of the graph to determine the sense of the target word that is in some way "optimal". The methods differ mainly

Distance	Synonyms	Siblings	Other
same sentence	1.0	1.0	1.0
1 sentence	1.0	1.0	1.0
2 sentences	1.0	0.5	0.3
3 sentences	1.0	0.5	0.3
next paragraph	0.5	0.3	0.2
farther	0.5	0.3	0.0

Table 3.1: Weighting scheme used in Galley and McKeown (2003). Weights are based on distance between the word instances in the text and on the type of relation: synonym, sibling (hyponyms of the same hypernym), or other (hypernym, hyponym, antonym, holonym and meronym).

in the amount of context they use (sentence or whole document), type of semantic relationships considered, and in their measure of "optimality".

3.2.2.1 Lexical Chains

Lexical cohesion is often represented via lexical chains, i.e., sequences of related words spanning a topical text unit (Morris and Hirst, 1991). Algorithms for computing lexical chains often perform WSD before inferring which words are semantically related. Here we describe one such disambiguation algorithm, proposed by Galley and McKeown (2003), while omitting the details of creating the lexical chains themselves.

Galley and McKeown's (2003) method consists of two stages. First, for each document, a graph is built representing all possible interpretations (senses) of the target words in question. Word-senses are nodes in the graph, and semantic relations are weighted edges. The text is processed sequentially, comparing each word against all words previously read. If a relation exists between the senses of the current word and any possible sense of a previous word, a connection (edge) is formed between the appropriate words and senses. The strength of the connection is a function of the type of relationship and of the distance between the words in the text (in terms of words, sentences and paragraphs). The set of relations being considered is a parameter that can be tuned experimentally. The original algorithm used the heuristic weighting scheme shown in Table 3.1, which is based on the type of WordNet relation between the word-senses, and the distance between them in the text.

In the disambiguation stage, all occurrences of a given word are collected together.

For each sense of a target word, the strength of all connections involving that sense are summed, giving that sense a unified score. The sense with the highest unified score is chosen as the correct sense for the target word. In subsequent stages the actual connections comprising the highest unified score are used as a basis for computing the lexical chains.

The algorithm is based on the 'one sense per discourse' hypothesis and groups together the information from every occurrence of the ambiguous target word in the document, in order to decide the appropriate sense. It is therefore a type-based algorithm, since it tries to determine the sense of the word in the entire document at once, and not separately for each instance.

3.2.2.2 Structural Semantic Interconnections

Inspired by lexical chains, Navigli and Velardi (2005) developed Structural Semantic Interconnections (SSI), a WSD algorithm which makes use of an extensive lexical knowledge base. This knowledge base is primarily based on WordNet and its standard relation set (i.e., hypernymy, meronymy, antonymy, similarity, nominalization and pertainymy) but is also enriched with collocation information representing semantic relatedness between sense pairs. Collocations are gathered from existing resources (such as the Oxford Collocations, the Longman Language Activator, and collocation web sites). Each collocation is mapped to the WordNet sense inventory in a semi-automatic manner (Navigli, 2005) and transformed into a *relatedness* edge.

Given a local word context $C = \{w_1, \ldots, w_n\}$ (the sentence containing the target word), SSI builds a graph $G = (V, E)$ such that $V = \bigcup_{i=1}^{n} senses(w_i)$ and an edge $(s, s') \in E$ exists if there is at least one interconnection between s (a sense of the word) and s' (a sense of its context) in the lexical knowledge base. The set of valid interconnections is determined by a manually-created context-free grammar consisting of a small number of rules. In effect, interconnections are paths comprised of one or more relations, connecting the two senses. Disambiguation is performed in an iterative fashion. First, a set I is created, containing the senses of words yet to be disambiguated. Initially, this set contains all senses of all words in the context. In each step, for each sense s of a word in I, SSI determines the degree of connectivity between s and the other senses in I:

$$SSIScore(s) = \frac{\sum_{s' \in I \setminus \{s\}} \sum_{j \in Interconn(s,s')} \frac{1}{length(j)}}{\sum_{s' \in I \setminus \{s\}} |Interconn(s,s')|} \tag{3.2}$$

where $Interconn(s,s')$ is the set of interconnections between senses s and s'. The contribution of a single interconnection is given by the reciprocal of its length, calculated as the number of edges connecting its ends. The overall connectivity score is then normalized by the number of contributing interconnections. The highest ranking sense s of word w_i is chosen and the senses of w_i are removed from the context I. The procedure terminates when either I is the empty set or there is no sense such that its *SSIScore* exceeds a fixed threshold.

3.2.2.3 Sequence Data Labeling

Mihalcea (2005a) uses a graph over word sequences, where the vertices are the different senses of each word, and are connected by edges to senses of the k previous words in the sequence (in the article, $k = 3$), in a fashion similar to Markov chains. The algorithm consists of two stages. In the first stage, the graph is created. A Lesk-like algorithm (see Section 3.2.1.1) over dictionary definitions is used in order to determine the weight of the edges between senses. In principle, the strength of the edges could be determined by any measure of sense similarity (for example, WordNet-based). The authors chose to use a dictionary-gloss-based similarity measure in order not to rely on the existence of a graph structure such as WordNet.

In the second stage, a score is associated with each node (word-sense). Starting with uniform scores for the nodes, an iterative procedure is applied to the graph, propagating the scores from each node to the next, based on the score itself, and on the weight of the connecting edge. For this purpose, the PageRank algorithm (Brin and Page, 1998) is used. Under certain conditions, which are fulfilled in this model, the algorithm is guaranteed to converge to a stationary state. The procedure concludes when convergence is reached. For each word, the sense with the highest score can then be determined.

3.2.2.4 Summary

As mentioned at the beginning of Section 3.2.2, all graph-based methods share a similar representation. However, they differ in several respects. The amount of context used varies from a single sentence (SSI) to the whole document (Lexical Chains), with

the sequence-labeling method falling somewhere between, since it considers the entire document as a sequence, but only uses relations (edges) between words that are close together in the text. The algorithms also differ with regard to whether they take a type- (Lexical Chains) or token-based approach (SSI and Sequence Labeling), and to the type of relations and weighting schemes they use. Finally, these different algorithms employ very different scoring methods to determine the "optimal" senses, ranging from a simple summation of edge weights (Lexical Chains), to an more complex scoring scheme with an iterative factor (SSI), to a sophisticated random-walk algorithm borrowed from the field of network analysis.

3.2.3 Vector-Based Models

3.2.3.1 Topic Tagging

Hearst and Schütze (1993) were among the first to consider a vector-based representation of word meaning. This representation was used to augment and rearrange an existing structured lexicon (WordNet), and to classify new words into existing categories. It is the latter task that is of interest here. It is not, strictly speaking, a WSD task, since words are assigned to general topics, rather than specific senses. However, the method presented in the paper was the foundation of many later vector-based WSD algorithms. In addition, topic-tags can be considered a form of coarse-grained sense inventory. In fact, the Macquire Thesaurus annotates word senses with similar category labels, and has been used as a sense inventory for WSD (e.g., Mohammad and Hirst 2006).

Hearst and Schütze's (1993) procedure works in two stages. In the first stage, WordNet is divided into sections representing topics. The second stage assigns proper names and new words (not in the existing lexicon) to one of these topics. Every word in the target corpus (five months of articles from the New York Times) is represented by a vector of co-occurrence counts. The cosine of the angle between the vectors of two words in this space is taken as a measure of the semantic similarity between the words. For each new word, the twenty nearest neighbors (most similar known words) are found. The target word is assigned to the category to which the largest number of neighbors belong. The algorithm was tested on proper nouns with a strong connection to a specific category, and was largely successful (only one clear error) in assigning them to their correct category. It was also evaluated on the 27 words from the test-set document that did not have an entry in WordNet. On these words, the results were

mixed, with 63% assigned to their correct category, 19% to related categories, and 19% incorrectly assigned.

3.2.3.2 Distributional and WordNet Similarity

McCarthy et al. (2004) propose a method for automatically ranking the senses of ambiguous words from raw text. Their approach is based on the methodology presented in Hearst and Schütze (1993), and stated more explicitly in Widdows (2003):

- For a unknown word, find 'corpus-derived neighbors' - words in the corpus whose occurrences are similar to the target.
- Map the target word to the place in the taxonomy where these neighbors are most concentrated.

In order to adapt this methodology to the task of sense ranking, some modification must be made. Instead of placing a new word in the taxonomy, we wish to determine which among the existing senses of the word is most appropriate to its use in the corpus. Therefore, sense ranking is equivalent to quantifying the degree of similarity among the neighbors and the different sense descriptions of the polysemous target word. Semantic similarity between words and senses within the taxonomy can be calculated using one of the many available WordNet similarity measures (see Section 2.3.1).

Let $N(w) = \{n_1, n_2, \ldots, n_k\}$ be the k most (distributionally) similar words to an ambiguous target word w, and $senses(w) = \{s_1, s_2, \ldots s_n\}$ the set of senses for w. For each sense s_i and for each neighbor n_j, the algorithm selects the neighbor's sense which has the highest WordNet similarity score (*wnss*) with regard to s_i.

$$wnss(s_i, n_j) = \max_{ns_x \in senses(n_j)} wnss(s_i, ns_x) \tag{3.3}$$

The ranking score of sense s_i is then increased as a function of this WordNet similarity score and the distributional similarity score (*dss*) between the target word and the neighbor:

$$RankScore(s_i) = \sum_{n_j \in N_w} dss(w, n_j) \frac{wnss(s_i, n_j)}{\sum_{s'_i \in senses(w)} wnss(s'_i, n_j)} \tag{3.4}$$

The predominant sense is simply the sense with the highest ranking score (*RankScore*) and can be consequently used to perform type-based disambiguation. The method presented above has four parameters: (a) the semantic space model representing the

distributional properties of the target words (it is acquired from a large corpus representative of the domain at hand and can be augmented with syntactic relations such as subject or object), (b) the measure of distributional similarity for discovering neighbors (see Lee 1999 for an overview), (c) the number of neighbors that the ranking score takes into account, and (d) the measure of sense similarity (see Budanitsky and Hirst 2001 for an overview of WordNet-based similarity measures).

3.3 Comparison of Unsupervised WSD Algorithms

3.3.1 Selection of Representative Algorithms

For our experiments, we chose four of the algorithms described in the previous section: Extended Gloss Overlap (Overlap), Lexical Chains (LexChains), Distributional and WordNet Similarity (Similarity) and Structural Semantic Interconnections (SSI). These were selected to represent the wide variety of approaches to unsupervised WSD. The methods differ in the type of representation they use, ranging from simple bag-of-words to various graph representations to statistical vector models. The methods can also be divided along other lines. The type of features and relations that they employ vary, from simple first order relations (word-overlap or first-order WordNet relations) to more complex statistical similarity and complex relationship paths in a graph. Another important issue is the division between type- and token-based approaches. Some of the methods described (Lexical Chains and Similarity) require the 'one-sense-per-discourse' assumption in order to obtain enough information about the ambiguous word. In fact, the Similarity approach makes an even greater simplification and uses one sense of each word for the entire corpus. The methods vary in the amount of data they employ for disambiguation. SSI and Overlap rely on sentence-level information for disambiguation, whereas Similarity and LexChains utilize the entire document or corpus. This enables the accumulation of large amounts of data regarding the ambiguous word, but does not allow separate consideration of each individual occurrence of that word. LexChains and Overlap take into account a restricted set of semantic relations (paths of length one) between any two words in the whole document, whereas SSI and Similarity use a wider set of relations.

To summarize, we selected representative models from the categories described in Section 3.2, which vary along the following dimensions: (a) the type of WSD performed (i.e., token-based vs. type-based), (b) the representation and size of the context

Method	WSD	Context	Relations
LexChains	types	document	first-order
Overlap	tokens	sentence	first-order
Similarity	types	corpus	higher-order
SSI	tokens	sentence	higher-order

Table 3.2: Properties of four WSD algorithms.

surrounding an ambiguous word (i.e., graph-based vs. word-based, document vs. sentence), and (c) the number and type of semantic relations considered for disambiguation. The properties of the selected WSD algorithms are shown in Table 3.2.

3.3.2 Experimental Setup

We compared the results of the four selected methods on two different, though closely related, tasks. The first task is finding the predominant sense of a polysemous word in the text. Both the similarity-based method and the lexical chains method were initially designed for this task. The Gloss Overlap and SSI methods were designed for the more specific WSD task, and need to be modified to the task of predominant sense detection. This adaptation was done by simply having the method find the correct sense of every occurrence of the target word in the text, and selecting the sense which was chosen most frequently.

The second task we addressed was the disambiguation of individual instances (tokens) in context, which is the most relevant in terms of application. This task is also more precise, and therefore the accuracy on this task is expected to be much lower than in the predominant sense detection task. In addition, only the Gloss Overlap and SSI algorithms are designed for this task. For the other algorithms, the only option is to tag all occurrences of each word with the predominant sense found for that word, and hope that this sense is strongly predominant, and covers a large portion of the individual instances. This technique of using the estimated predominant sense to label all instances, disregarding context, can also be applied to Gloss Overlap and SSI algorithms, in the following manner. First, token-based disambiguation is performed on all the ambiguous instances. Then, the most frequently chosen sense-label for each word type is determined, and used to re-label all the instances of that word. For Gloss Overlap and SSI, both the direct labeling and the predominant-sense technique were tested.

Our experiments were conducted on the SemCor corpus, on the same 2,595 polysemous nouns (53,674 tokens) used as a test set by McCarthy et al. (2004). These nouns were those which occurred more than twice in SemCor and more than ten times in the British National Corpus (BNC). Our experiments use the WordNet (version 1.7.1) sense inventory. However, the approaches are not limited to this particular lexicon and can be adapted for use with other resources with traditional dictionary-like sense definitions and alternative structure.

The following notation describes our evaluation measures: W is the set of all ambiguous noun types in the SemCor corpus ($|W| = 2,595$). $Senses(w)$ is the set of senses for noun type w, while $f_g(w)$ and $f_m(w)$ refer to w's most frequent (predominant) sense according to the SemCor gold standard and our algorithms, respectively. Finally, $T(w)$ is the set of tokens of w and $sense_g(t)$ denotes the sense assigned to token t according to SemCor.

We first measure how well our algorithms can identify the predominant sense, if one exists:

$$Acc_{ps} = \frac{|\{w \in W \,|\, f_m(w) = f_g(w)\}|}{|W|} \tag{3.5}$$

A baseline for this task can be easily defined for each word type by selecting a sense at random from its sense inventory and assuming that this is the predominant sense:

$$Baseline_{sr} = \frac{1}{|W|} \sum_{w \in W} \frac{1}{|senses(w)|} \tag{3.6}$$

We evaluate the algorithms' token-based disambiguation performance, using the detected predominant sense $f_m(w)$ to label all tokens, by measuring the ratio of tokens for which our models choose the right sense:

$$Acc_{wsd/ps} = \frac{\sum_{w \in W} |\{t \in T(w) | f_m(w) = sense_g(t)\}|}{\sum_{w \in W} |T(w)|} \tag{3.7}$$

In the predominant sense detection task, in case of ties in SemCor, any one of the predominant senses was considered correct. Also, all algorithms were designed to randomly choose from among the top scoring options in case of a tie in the calculated scores. This introduces a small amount of randomness (less than 0.5%) in the accuracy calculation, and was done to avoid the pitfall of defaulting to the first sense

listed in WordNet, which is usually the actual predominant sense in SemCor (the order of senses in WordNet is based primarily on the SemCor sense distribution) and thus overestimating accuracy.

In all the experiments in this chapter, we used the χ^2-test to evaluate statistical significance. Unless otherwise stated, we took $p < 0.01$ to indicate a statistically significant difference.

3.3.3 Parameter Settings

We did not specifically tune the parameters of our WSD algorithms on the SemCor corpus, as our goal was to use hand labeled data solely for testing purposes. We selected parameters that have been considered "optimal" in the literature, although admittedly some performance gains could be expected had parameter optimization taken place.

For Overlap, we used the semantic relations proposed by Banerjee and Pedersen (2003), namely hypernyms, hyponyms, meronyms, holonyms, and troponym synsets. We also adopted their overlap scoring mechanism which treats each gloss as a string of words and assigns an n-word overlap the score of n^2. Function words were not considered in the overlap computation. For LexChains, we used the relations reported in Galley and McKeown (2003). These are all first-order WordNet relations, with the addition of the *siblings* – two words are considered siblings if they are both hyponyms of the same hypernym. The relations have different weights, depending on their type and the distance between the words in the text (see Table 3.1). These weights were imported from Galley and McKeown (2003) into our implementation without modification.

Because the SemCor corpus is relatively small (less than 700,00 words), it is not ideal for constructing a neighbor thesaurus appropriate for McCarthy et al.'s (2004) method. This method requires each word to participate in a large number of co-occurring contexts in order to obtain reliable distributional information. To overcome this problem, we followed McCarthy et al. and extracted the neighbor thesaurus from the entire BNC. We also recreated their semantic space, using a RASP-parsed (Briscoe and Carroll, 2002) version of the BNC and their set of dependencies (i.e., Verb-Object, Verb-Subject, Noun-Noun and Adjective-Noun relations). Similarly to McCarthy et al., we used Lin's (1998b) measure of distributional similarity, and considered only the 50 highest ranked neighbors for a given target word. Sense similarity was computed using the Lesk's (Banerjee and Pedersen, 2003) WordNet similarity

Method	Acc_{ps}	$Acc_{wsd/dir}$	$Acc_{wsd/ps}$
UpperBnd	100%	–	68.4%
SSI	53.7%	42.7%	47.9%
Similarity	54.9%	–	46.5%
Overlap	49.4%	36.5%	42.5%
LexChains	48.3%	–	40.7%
Baseline	34.5%	–	23.0%

Table 3.3: Results of individual disambiguation algorithms on SemCor nouns[2]. Scores represent accuracy on three tasks: predominant sense detection (Acc_{ps}), context-specific WSD ($Acc_{wsd/dir}$), and WSD using the automatically acquired predominant sense to label all instances ($Acc_{wsd/ps}$).

measure[1].

3.3.4 Results

The performance of the individual algorithms is shown in Table 3.3. We also include the random-sense baseline discussed in Section 3.3 and the upper bound of defaulting to the actual first (i.e., most frequent) sense provided by the manually annotated SemCor. We report predominant sense accuracy (Acc_{ps}), and WSD accuracy when using the automatically acquired predominant sense ($Acc_{wsd/ps}$) to label all instances of the word. For token-based algorithms, we also report their WSD performance in context, i.e., without use of the predominant sense ($Acc_{wsd/dir}$).

As expected, the accuracy scores in the WSD task are lower than the respective scores in the predominant sense task, since detecting the predominant sense correctly only insures the correct tagging of the instances of the word with that first sense. All methods perform better than the baseline in the predominant sense detection task, and the difference is statistically significant. LexChains and Overlap perform significantly worse than Similarity and SSI, whereas LexChains is not significantly different from

[1]This measure is identical to our Extended Gloss Overlap from Section 3.2.1.1, but instead of searching for overlap between an extended gloss and a word's context, the comparison is done between the two extended glosses of two synsets.

[2]The LexChains results presented here are not directly comparable to those reported by Galley and McKeown (2003), since they tested on a subset of SemCor, and included monosemous nouns. They also used the first sense in SemCor in case of ties. The results for the Similarity method are slightly better than those reported by McCarthy et al. (2004) due to minor improvements in implementation.

	Overlap	LexChains	Similarity
SSI	30.48%	31.67%	37.14%
Similarity	35.87%	33.10%	
LexChains	28.05%		

Table 3.4: Portion of words for which each pair of algorithms correctly assigned the predominant sense (as % of all words).

Overlap. Likewise, the difference in performance between SSI and Similarity is not significant. With respect to WSD, all the differences in performance are statistically significant.

Interestingly, for the Overlap and the SSI algorithms, using the detected predominant sense to tag all instances is preferable to tagging each instance individually (compare $\text{Acc}_{wsd/dir}$ and $\text{Acc}_{wsd/ps}$ for these algorithms in Table 3.3). This means that a large part of the instances which were not tagged individually with the predominant sense were actually that sense.

As we can see from the last line of Table 3.3, on average, the most frequent sense accounts for 68% of word occurrences. It is interesting to note that for all three methods, the disambiguation score using the calculated first sense is approximately 85% of the predominant sense accuracy. This implies that the methods are more successful on more frequent words or on those with a strong skew to the predominant sense (leading to higher ratio than the expected 68%).

3.4 Ensembles for WSD

3.4.1 Motivation

A close examination of the performance of the individual methods in the predominant sense detection task shows that while the accuracy of all the methods is within a range of 7%, the actual words for which each algorithm gives the correct predominant sense are very different. Table 3.4 shows the degree of overlap in assigning the appropriate predominant sense among the four methods. As can be seen, the largest amount of overlap is between Similarity and SSI, and this corresponds to approximately two thirds the words they correctly label. This means that each of these two methods correctly assigns the predominant sense to more than 350 words which the other labels

incorrectly.

If we had an "oracle" which would tell us which method to choose for each word, we would achieve approximately 82.4% in the predominant sense task, giving us 58% in the WSD task. We see that there is a large amount of complementation between the algorithms, where the successes of one make up for the failures of the others. This suggests that the errors of the individual methods are sufficiently uncorrelated, and that some advantage can be gained by combining their predictions. These observations, along with the differences between the methods and the variety of the information sources and interactions they use (as described in Section 3.3.1 and summarized in Table 3.2), lead us to the next set of experiments, which investigate the unsupervised combination of WSD algorithms.

3.4.2 Ensemble Methods

An important finding in machine learning is that a set of classifiers whose individual decisions are combined in some way (an *ensemble*) can be more accurate than any of its component classifiers, provided that the individual components are relatively accurate and diverse (Dietterich, 1997). This simple idea has been applied to a variety of classification problems, ranging from optical character recognition to medical diagnosis, part-of-speech tagging (see Dietterich 1997 and van Halteren et al. 2001 for overviews), and notably supervised WSD (Florian et al., 2002).

Since our effort is focused exclusively on unsupervised methods, we cannot use most machine learning approaches for creating an ensemble (e.g., stacking), as they require a labeled training set. We therefore examined several basic ensemble combination approaches that do not require training data for parameter estimation.

3.4.3 Formulation

We define $Score(M_i, s_j)$ as the (normalized) score which a method M_i gives to word sense s_j. The predominant sense calculated by method M_i for word w is then determined by:

$$PS(M_i, w) = \underset{s_j \in senses(w)}{\text{argmax}} \; Score(M_i, s_j) \tag{3.8}$$

All ensemble methods receive a set $\{M_i\}_{i=1}^{k}$ of individual methods to combine, so we denote each ensemble method by $MethodName(\{M_i\}_{i=1}^{k})$.

Direct Voting Each ensemble component has one vote for the predominant sense, and the sense with the most votes is chosen. The scoring function for the voting ensemble is defined as:

$$Score(Voting(\{M_i\}_{i=1}^{k}), s)) = \sum_{i=1}^{k} eq[s, PS(M_i, w)] \tag{3.9}$$

$$\text{where } eq[s, PS(M_i, w)] = \begin{cases} 1 & \text{if } s = PS(M_i, w) \\ 0 & \text{otherwise} \end{cases}$$

Probability Mixture Each method provides a probability distribution over the senses. These probabilities (normalized scores) are summed, and the sense with the highest score is chosen:

$$Score(ProbMix(\{M_i\}_{i=1}^{k}), s)) = \sum_{i=1}^{k} Score(M_i, s) \tag{3.10}$$

Rank-Based Combination Each method provides a ranking of the senses for a given target word. For each sense, its placements according to each of the methods are summed and the sense with the lowest total placement (closest to first place) wins.

$$Score(Ranking(\{M_i\}_{i=1}^{k}), s)) = \sum_{i=1}^{k} (-1) \cdot Place_i(s) \tag{3.11}$$

where $Place_i(s)$ is the number of distinct scores that are larger or equal to $Score(M_i, s)$.

Arbiter-based Combination An alternative to ensemble methods where each member plays an equal part is arbiter-base combination. One WSD method can act as an arbiter for adjudicating disagreements among component systems. It makes sense for the adjudicator to have reasonable performance on its own. We therefore selected SSI as the arbiter since it had the best accuracy on the WSD task (see Table 3.3).

For each disagreed word w, and for each sense s of w assigned by any of the systems in the ensemble $\{M_i\}_{i=1}^{k}$, we calculate the following score:

$$Score(Arbiter(\{M_i\}_{i=1}^{k}), s) = SSIScore^*(s) \tag{3.12}$$

where $SSIScore^*(s)$ is a modified version of the score introduced in Section 3.2.2.2 which excludes from the context used by SSI the senses of w which were not chosen by any of the systems in the ensemble. Therefore, the context used for w is the set of agreed senses and the remaining words of each sentence. This effectively reduces the number of possibilities considered by the arbiter and can positively influence the

algorithm's performance, since it eliminates noise coming from senses which are likely to be wrong.

An example of the way the various ensembles work is given in Table 3.5. For this example we chose the the word *'sense'*, which has five senses in WordNet (for definitions of the senses and further details, see Section 4.3.2). The top portion of the table gives the original scores assigned by each of the members in our ensemble to each of the senses. As can be seen, the scoring systems vary considerably between the different algorithms, and therefore can not be compared directly. The Probability Mixture method partially addresses this problem by normalizing the scores to create probability distributions. These can be summed to determine the most probable sense taking into consideration the scores given by all the ensemble members. The Voting ensemble uses only one piece of information from each member - the identity of the most likely sense according to that algorithm. It does not take into consideration the scores for the other senses. This approach has several advantages. Firstly, because there is no standardized scoring system, the meaning of the scores is unclear. By only considering the highest-scoring sense, this ensemble method is, in a sense, leveling the field, and forcing all the members to use the same criterion. Secondly, the Voting ensemble is the only one that can be used naturally with WSD methods that do not use a scoring system, but only give the estimated correct sense. On the other hand, when scores are available, the Voting ensemble throws away a lot of potentially useful information. The Ranking ensemble tries to get the best of both worlds. It asks the methods to rank the various senses, and in this way transforms the different scoring mechanisms to a standard ranking system. It retains a large part of the information implicit in the scores, although it dispenses with the exact score values. It can, therefore, take into account the opinion of each ensemble method on all of the senses, not just the highest-scoring one. It can also be used with WSD methods that do not provide scores, under the assumption that the sense estimated as correct by the algorithm is ranked first, and all the rest share second place.

3.4.4 Method and Parameter Settings

We assess the performance of the different ensemble systems on the same set of SemCor nouns on which the individual methods were tested. For the best ensemble, we also report results for disambiguating all-nouns in the Senseval-3 all-words task, in order to compare with state-of-the-art WSD algorithms. The parameter settings for the

		Sense 1	Sense 2	Sense 3	Sense 4	Sense 5
Original Scores	SSI	16	41	30	0	3
	Similarity	0.175	0.473	0.303	0.174	0.226
	Overlap	22	179	32	61	19
	LexChains	458.6	514.5	571.7	444.9	417.2
ProbMix	SSI	0.18	0.46	0.33	0.00	0.03
	Similarity	0.13	0.35	0.22	0.13	0.17
	Overlap	0.07	0.57	0.10	0.20	0.06
	LexChains	0.19	0.21	0.24	0.19	0.17
	Ensemble	0.57	**1.59**	0.89	0.52	0.43
Voting	SSI	–	vote	–	–	–
	Similarity	–	vote	–	–	–
	Overlap	–	vote	–	–	–
	LexChains	–	–	vote	–	–
	Ensemble	0	**3**	1	0	0
Ranking	SSI	3	1	2	5	4
	Similarity	4	1	2	5	3
	Overlap	4	1	3	2	5
	LexChains	3	2	1	4	5
	Ensemble	14	**5**	8	16	17

Table 3.5: Example of the operation of the different ensemble methods on the word *sense*. Tables show the contribution of each component in the ensemble to each sense under the different ensemble setups, and the resulting scores for the ensemble as a whole. Winning scores are denoted in **bold**.

Method	Acc_{ps}	$\text{Acc}_{wsd/ps}$
UpperBnd	100%	68.4%
Rank-based	58.1%	50.3%
Voting	57.3%	49.8%
PrMixture	57.2%	50.4%
Arbiter-based	56.3%	48.7%
Similarity	54.9%	46.5%
SSI	53.5%	47.9%

Table 3.6: Accuracy for ensemble combinations on predominant sense detection (Acc_{ps}) and WSD using the detected predominant sense ($Acc_{wsd/ps}$).

individual members of our ensembles are the same as described in Section 3.3.3.

3.4.5 Results

Our results are summarized in Table 3.6. All ensemble methods perform significantly better than the best individual methods, i.e., Similarity and SSI. On the WSD task, the voting, probability mixture and rank-based ensembles significantly outperform the arbiter-based one. The performances of the probability mixture and rank-based combinations do not differ significantly, but both ensembles are significantly better than voting. One of the factors contributing to the arbiter's worse performance (compared to the other ensembles) is the fact that in many cases (almost 30%), none of the senses suggested by the other methods was correct. In these cases, there is no way for the arbiter to select the correct sense.

We also examined the relative contribution of each component to overall performance. Table 3.7 displays the drop in performance by eliminating any single component from the rank-based ensemble (indicated by $-$). The system that contributes the most to the ensemble is SSI, which is also the best performing individual system, for context-specific WSD (see Table 3.3). Interestingly, the removal of Overlap and Similarity cause a similar decrease in WSD accuracy (0.6 and 0.9, respectively), despite the difference between them in individual performance. However, the effect of their removal on the predominant sense accuracy, is very different (0.5 for Overlap, compared to 1.8 for Similarity). This seems to indicate that the decrease in WSD accuracy stems from different sources in the two algorithms. The Similarity method performs weakly

Ensemble	Acc_{ps}	$\text{Acc}_{wsd/ps}$
–Rank-based	58.1%	50.3%
–Overlap	57.6% (−0.5%)	49.7% (−0.6%)
–LexChains	57.2% (−0.7%)	50.2% (−0.1%)
–Similarity	56.3% (−1.8%)	49.4% (−0.9%)
–SSI	56.3% (−1.8%)	48.2% (−2.1%)

Table 3.7: Decrease in accuracy as a result of removal of each method from the rank-based ensemble.

Method	Acc_{ps}	$\text{Acc}_{wsd/dir}$	$\text{Acc}_{wsd/ps}$
UpperBnd	63.10%	–	61.10% (68.72%)
Rank-Based	56.56%	–	55.01% (63.89%)
SSI	55.76%	50.26% (59.97%)	53.39% (62.52%)
Similarity	45.84%	–	46.88% (57.28%)
Baseline	29.50%	–	21.50% (36.80%)

Table 3.8: Results of individual disambiguation algorithms on Senseval 3 nouns. Results in parentheses include monosemous nouns.

on frequent words, and strongly on rarer ones (as shown in Figure 3.1 and discussed below). Its removal would therefore effect the ensemble's accuracy mostly on infrequent words, which translates to a smaller effect when counting actual instances. Overlap, on the other hand, has similar performance in all frequency bands, and therefore a similar effect on both predominant-sense accuracy, and per-instance WSD.

Figure 3.1 shows the WSD accuracy of the best single methods and the ensembles as a function of the noun frequency in SemCor. We can see that there is at least one ensemble outperforming any single method in every frequency band, and that the rank-based ensemble consistently outperforms Similarity and SSI in all bands. Although Similarity has an advantage over SSI for low and medium frequency words, it delivers worse performance for high frequency words. This is possibly due to the quality of neighbors obtained for very frequent words, which are not semantically distinct enough to reliably discriminate between different senses. This factor also strongly affects the results of the voting combination, though it has less impact on the ranking and probability mixture combination, which take into account the scores of all the senses.

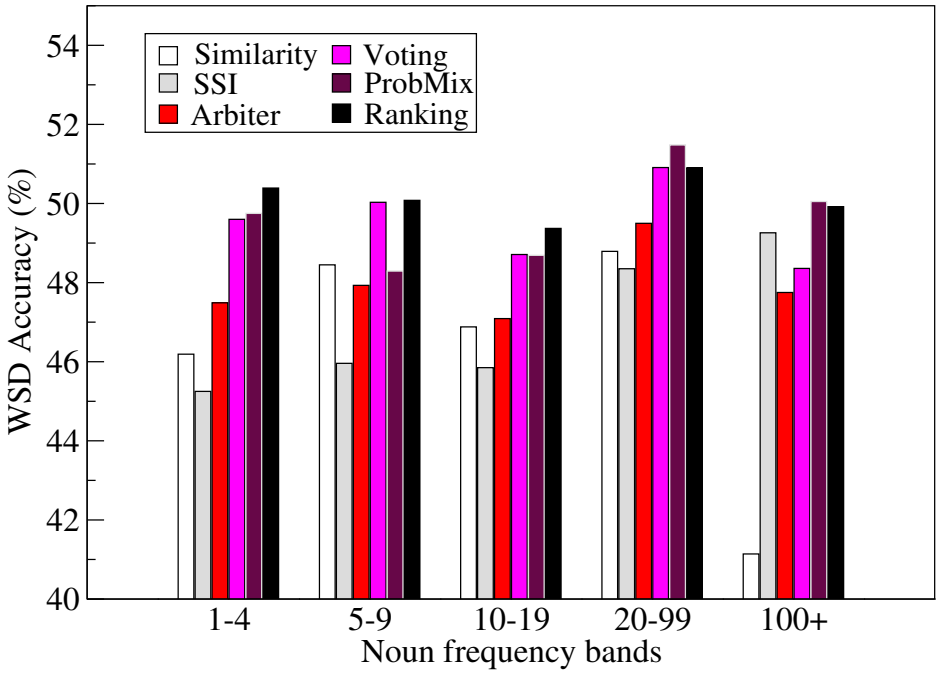

Figure 3.1: WSD accuracy as a function of noun frequency in SemCor, for Similarity, SSI, and four ensembles.

Table 3.8 lists the performance of the rank-based ensemble on the Senseval 3 corpus. We also report results for the best individual methods, namely SSI and Similarity. Our baseline selects the first sense randomly and uses it to disambiguate all instances of a target word. Our upper bound defaults to the first sense from SemCor (UpperBnd). For the WSD task, aside from the accuracy on polysemous nouns, which is comparable to our evaluation and results on SemCor, we also report in parentheses the overall accuracy on nouns, including monosemous ones. These numbers are comparable to the official scores reported for the Senseval 3 task. The best unsupervised system that participated in Senseval 3 was IRST-DDD, and achieved 61.2% F-Score. This system was developed by Strapparava et al. (2004) and performs domain driven disambiguation (IRST-DDD). Specifically, the approach compares the domain of the context surrounding the target word with the domains of its senses and uses a version of WordNet augmented with domain labels (e.g., economy, geography).

Our rank-based ensemble outperforms Similarity and SSI as well as the IRST-DDD system. This is an encouraging result, suggesting that there may be advantages in developing diverse classes of unsupervised WSD algorithms for system combination. We further note that not all of the components in our ensemble are optimal. Predominant senses for Lesk and LexChains were estimated from the Senseval 3 data, however, a

larger corpus would probably yield more reliable estimates. Also, one would expect better performance with better disambiguators (e.g., by substituting Lesk or LexChains with IRST-DDD).

3.5 Discussion

Comparison of Existing Approaches As stated in the beginning of this chapter, one of our goals was to perform a detailed comparison of existing WSD approaches under a common framework. Our experiments revealed some important insights. The Overlap and LexChains methods, which employed relatively simple representation and scoring schemes, perform poorly compared to the other methods, and do not come close to state-of-the-art results. Though simplicity is desirable, it can not come at the expense of performance. SSI has the best performance among the methods we compared. However, it makes use of extensive lexical knowledge, much of which was laboriously collected from various linguistic resources. It also uses a complex representation, which relies heavily on the structure of WordNet. These resources do not exist in most other languages, and are difficult to adapt and extend to new domains. This makes SSI problematic as a basis for a wide-coverage, general purpose, WSD algorithm. The last of the algorithms we examined, Similarity, achieves good performance while requiring only a small amount of lexical knowledge. Its vector-based representation is relatively simple, and has the added advantage of being a standard representation employed by many supervised algorithms. This means that tools or ideas borrowed from the supervised framework (like the ensemble idea used in this chapter) are more likely to work well with this representation. The dependence of the method on corpus statistics rather than lexical knowledge makes it easy to port to new domains and other languages, provided sufficient (unannotated) text is available.

Predominant Sense One of the insights gained from our comparison has to do with the usefulness and strength of using the automatically estimated predominant sense. Our experiments have shown that using the predominant sense to tag all instances can often outperform true token-based WSD methods. However, such an approach is not a feasible disambiguation technique, since completely ignoring secondary senses is unacceptable in most real-world situations where WSD is required. Nonetheless, there is much to be gained from the strength of the predominant-sense. We now know that the automatically-estimated predominant sense can be used as a fallback option in

cases of uncertainty or insufficient contextual information, thus increasing recall with little chance of losing precision.

Choice of Ensembles As stated in Section 3.4.2, our choice of ensembles was motivated by the need for simple methods which have minimal requirements from the underlying WSD algorithms. This was desirable in order to make the ensembles as widely applicable as possible. For this reason we chose ensembles which require only that a score be assigned to each possible sense by each of the component WSD algorithms. Our ensembles present several simple ways of making the most of these raw numbers. Using more sophisticated methods, we can expect better results, but at the cost of imposing restrictions on the underlying WSD algorithms, or requiring additional data (e.g., a small amount of manually annotated data in order to learn an optimal weighting of the component algorithms). In Section 6.3 we discuss possible improvements and modifications that can be made to the simple ensembles presented in this chapter.

Pros and Cons of Ensembles In this chapter we showed that simple ensemble techniques borrowed from the supervised framework can improve the results of existing unsupervised WSD methods without the extensive labor and system design needed for building an algorithm from scratch. Ensemble methods are especially suited for use with existing unsupervised WSD methods, since they turn one of the main disadvantages of existing methods – their lack of a standard representation, formulation and approach to the problem – into an advantage. They leverage the diversity and complementary nature of the different algorithms to help compensate for each other's weaknesses. In Section 6.2, we expand further on the type of scenario in which ensembles are most useful. Combination approaches have several drawbacks, however. They depend on the existence of a number of sufficiently diverse WSD systems, that can be run on similar input, and provide output according to a single inventory. Unless this setting already exists, it is doubtful whether creating it is worthwhile if the only aim is to increase accuracy, since the improvements resulting from the ensemble methods are relatively small. It is likely that, in such cases, investing the required time and effort in building a system from scratch would be a better choice.

All-Words vs. Representative Samples Some classic unsupervised WSD algorithms take the *all-words* approach, making use of linguistic connections between all

the words in the document. For instance, as described in Section 3.2.2, the Lexical Chains approach makes use of discourse connections, while the Sequence-Labelling approach makes use of both semantic connections and contextual connection between the words in a graphical model. Most of these approaches assume a lexical resource that provides information about every word and its relations to every other one.

On the other hand, the *lexical-sample* approach (so called because of the means by which it is evaluated, see Section 2.3.3.3), addresses each word individually as a separate disambiguation problem. From a machine learning perspective, this approach better suits the supervised setup we are trying to emulate. In the supervised setting, the common methodology is to train a classifier on each word individually, given the labeled data for that word. This breaks the problem down into a collection of stand-alone classification tasks, for which there exist many well-known machine learning solutions. From a pragmatic perspective, taking into account the type of applications and tasks for which unsupervised WSD is most useful, i.e., new domains, the single-word approach has a further advantage. It is more adaptable, and allows the extension and modification of a small part of the lexicon while leaving the rest intact. This means that new words, or new senses of existing words, that are specific to the new domain can be augmented to an existing WSD system with little cost. It also allows different techniques to be used for some of the words. For instance, supervised methods can be used for words for which much labeled data is available, or for word classes where unsupervised methods show a weakness, such as highly polysemous or very infrequent words, while unsupervised methods address the rest of the words. For these reasons, the individual-word approach is most suited for our purposes in this thesis.

3.5.1 Summary

In this chapter we presented an evaluation study of four algorithms, representing four well-known approaches to unsupervised WSD. Our comparison involved type- and token-based disambiguation algorithms relying on different kinds of WordNet relations and different amounts of corpus data. Our experiments revealed two important findings. First, type-based disambiguation can yield results superior to a token-based approach. In other words, using the predominant sense is more accurate than disambiguating instances individually, even for token-based algorithms. Second, the outputs of the different approaches we examined are sufficiently diverse to motivate combination methods for unsupervised WSD. We defined several unsupervised ensembles

which combine the predominant sense outputs of individual methods, and showed that the combined systems outperformed their best component algorithms both on the SemCor and Senseval 3 data sets.

The issues discussed above point the way to our next step. In order to provide an accurate, reliable WSD system which can be adapted to new domains and other languages, we can not rely on existing methods, but must develop our own. Our approach should use a vector-based representation, and preferably employ corpus-based distributional similarity metrics. It should make use of ideas from supervised methods, and treat each word on an individual basis. Our method should be simple, and rely as little as possible on lexical resources. Finally, it should make use of the estimated predominant sense in cases of uncertainty, since our experiments have shown this to be a low-risk strategy for increasing recall.

Chapter 4

Automatic Creation of Sense-Labeled Training Data

4.1 Introduction

In the previous chapter we examined and compared a variety of existing unsupervised methods representing different approaches to the WSD problem. We noted the lack of a standard methodology and representation such as is common in the supervised framework, and suggested that this contributes to the accuracy gap between unsupervised and supervised methods. We presented ways to leverage the diversity of existing approaches to improve WSD results, using ensemble methods borrowed from the supervised learning setup. Our detailed examination of WSD approaches highlighted the importance of defaulting to the predominant sense in case of uncertainty, and the effectiveness of the distributional similarity approach to WSD. Our experiments also showed the potential benefits in borrowing ideas from the supervised methodology. However, they also served to emphasize the disadvantages inherent in having to rely on existing methods. Since these methods were designed independently of each other, and without regard for a wider learning framework, using them as the core of a WSD system does not result in an optimal setting. Our goal is to create a WSD system which will be free of human annotation, but can provide accuracy at a level comparable to state-of-the-art supervised methods. Our system should take into consideration the lessons we learned in the previous chapter, regarding the importance of the predominant sense and the potential of distributional similarity metrics. In order to accomplish this goal, it is necessary to design a system from the ground up, taking these factors into account from beginning to end.

We therefore developed an unsupervised WSD method which circumvents the question of actual disambiguation method, which is the main source of discrepancy in unsupervised WSD, and deals directly with the data. It automatically creates labeled training data, suitable for use with standard supervised classifiers. Our approach uses distributional similarity to find words which are similar to the target ambiguous word (distributional neighbors). It then associates each neighbor with a sense, through a semantic similarity measure. Sentences containing the neighbors are extracted from a large corpus, and the neighbors are replaced with instances of the target word, labeled with the sense associated with that neighbor. This procedure produces a labeled training dataset in a completely unsupervised fashion. The dataset can be used to train any standard supervised classifier which in turn can be used for disambiguation of the test data. Our approach shifts to the supervised setting before the disambiguation stage, thereby taking better advantage of the benefits supervised learning presents, such as a standard representation, and a selection of powerful, well studied, classifiers. We train several classifiers, based on a variety of learning paradigms, on our automatically-constructed dataset, and use them for disambiguation. The results are compared to those of the same classifiers, trained on manually-labeled data, and to other unsupervised WSD algorithms. Classifiers trained with our method significantly outperform those using other methods of data generation, and represent a big step in bridging the accuracy gap between supervised and unsupervised methods.

4.2 Related Work

As mentioned in the thesis introduction (Chapter 1), the problem of obtaining sufficient labeled data for supervised methods (the *data acquisition bottleneck*) is a major setback to useful and accurate word-sense disambiguation. Classical approaches to unsupervised WSD usually deal with the lack of labeled data by using other sources of information, such as lexical resources and linguistic knowledge. Some methods (e.g., the Lexical Chains algorithm, see Section 3.2.2) rely on semantic relations between words, provided by linguistic resources such as WordNet. Others make use of manually provided linguistic data (e.g., the collocation information used in the SSI algorithm described there). In the previous chapter we mentioned some of the problems with such approaches, notably the lack of a standardized formulation of the problem, and incompatibility of difference approaches. There are few methods which address the lack of labeled data directly, and these fall into two main categories: (1) extending

or expanding a small existing dataset, using bootstrapping techniques, and (2) automatically creating labeled data.

4.2.1 Bootstrapping Approaches

Yarowsky (1995) introduced one of the earliest semi-supervised algorithms for WSD. The algorithm uses a small seed group of hand-labeled instances or collocation features as a starting point in an iterative procedure. In each iteration, a classifier is trained on the labeled data, and used to tag the unlabeled data. Instances which are tagged with high confidence (scored above a certain threshold), are added to the labeled data. The algorithm also makes use of the *one-sense-per-discourse* heuristic. That is, if several instances in a single discourse are tagged with a single sense, the algorithm tags the rest of the instances with the same sense. This procedure is repeated, and in each stage, only tags scoring above the confidence threshold are retained. This allows the algorithm to overcome misclassifications that occur in previous stages. The algorithm makes use of the *supervised* Decision List method for WSD. The Decision List method takes into account a wide range of potential evidence sources (lemmas, parts-of-speech, inflected forms etc.) that co-occur with the ambiguous word (local or distant co-occurrences, or dependency relations) and may provide information about the word sense. These indicative features are ranked by their log-likelihood ratio score: $log\frac{p(sense\ A|feature)}{p(sense\ B|feature)}$ according to the training data (the algorithm considers only two main senses for each word). In the end, the algorithm creates a decision list based on the features, starting from the highest rank. The decision list is simply a set of *if-else* rules, stating that *"if feature X occurs, label with sense A, otherwise, proceed to next rule"*.

This procedure was applied to a small set of twelve nouns (due to the effort required to hand-label the seed group). For each word, distinctions were made between only two frequent but highly distinct senses, such as the *living-organism* and *factory* senses of *plant*, and the *bird* and *machine* senses of *crane*. In this restricted setting, the algorithm achieved high accuracy (96.1% average accuracy, with a most-frequent-sense baseline of 63.9%).

This method is of interest because of the way it essentially creates a labeled dataset which did not exist previously. It then makes use of the strengths of supervised methods to take advantage of this new data in a robust way. This bears some resemblance to the method we propose in this work.

Karov and Edelman (1996) describe an iterative method for WSD using a corpus and a machine-readable dictionary. They create a seed set of tagged examples by using words that appear uniquely in the definition of each of the senses. For each such related word, all sentences in the corpus which contain that word are labeled with the associated sense. The algorithm then proceeds in an iterative fashion - each instance (sentence containing the target word) in the corpus is labeled with the sense of the most similar sentence in the seed set. Similarity between sentences is measured by similarity of the words comprising them, and this, in turn, is measured by similarity of the sentences in which the words appear throughout the corpus. The circularity of the definitions is part of the iterative convergeant nature of the algorithm. In each iteration, the results of the previous iteration are used to calculate the two similarities (between sentences and between words), until the process converges. The algorithm was tested on a set of four nouns (*drug, sentence, suit,* and *player*), to disambiguate between two quite distinct senses. It achieved results similar to those of Yarowsky (1995), with accuracy ranging from 90.5% to 94.8%, while using a much smaller, automatically created, seed group.

This approach contains some elements similar to Yarowsky's (1995) work and our own. Both Yarowsky (1995) and Karov and Edelman (1996) use seeds, although the latter construct the seed set automatically, using the semantic resource (the dictionary), thereby making the method completely unsupervised. The method we present in this chapter also makes use of an automatically constructed dataset, but the method of construction is very different, and our procedure is not an iterative, bootstrapping, one. We create the dataset as a whole, rather than "growing" is from a seed set. Also, the notion of similarity discussed in the article is quite different from the one we employ. Karov and Edelman (1996) measure similarity between sentences, and use the sentence information to compare words. Our method makes use of distributional and semantic similarity at the individual words level, and does not consider sentences at all.

4.2.2 Unsupervised Creation of Labeled Data

Gale et al. (1992a) pioneered the use of parallel corpora as a source of sense-tagged data. Their key insight is that different translations of an ambiguous word can serve to distinguish its senses. Ng et al. (2003) extend this approach further and demonstrate that it is feasible for large scale WSD, and can achieve results comparable to those

of the systems that participated in Senseval 2. They gather examples from English-Chinese parallel corpora and use automatic word alignment as a means of obtaining a translation dictionary. Translations are next assigned to senses of English ambiguous words. English instances corresponding to these translations serve as training data. This approach provides a useful method for automatically creating training data for WSD, and is especially suited for disambiguation as part of a translation system. However, there are several drawbacks with this approach. First and foremost, it relies on multi-lingual parallel corpora, which have limited availability. While more easily obtained than manual sense-labeled data, they are still relatively rare, and there is no guarantee that one will be available in the domain of interest. The approach also depends on automatic word alignment methods, and therefore suffers if these methods are inaccurate. The translation-based approach is restricted in terms of the senses it can learn to distinguish, since only senses with different translations can be disambiguated. Finally, it is not entirely free from manual intervention. In order to be used as a general WSD system, a manual mapping must be performed between translations and senses.

Another way to automatically obtain sense-labeled data is to use related words from a dictionary to learn contextual cues for WSD (Mihalcea, 2002). Perhaps the first incarnation of this idea is found in Leacock et al. (1998), who describe a system for acquiring topical contexts that can be used to distinguish between senses of an ambiguous word. For each sense, related monosemous words are extracted from WordNet by making use of the various relationship connections between sense entries (i.e., hyponymy, hypernymy etc.). The system then queries the Web using these related words. The contexts retrieved for the monosemous words related to a specific sense are presumed to be indicators of that sense, and are used as training examples. A probabilistic Bayesian classifier, which takes into account both topical and local features, is trained on the retrieved examples. It is then used to disambiguate occurrences of the target word.

The authors evaluated the effectiveness of their method on a manually annotated corpus comprised of instances of one noun (*line*), one verb (*serve*) and one adjective (*hard*). They examined and tried to draw conclusions regarding the effects of using topical vs. local features for each of these words, as representatives of their respective parts-of-speech. A similar idea, proposed by Yarowsky (1992), is to use a thesaurus and acquire informative contexts from words in the same category as the target.

Such methods do not rely on parallel corpora, and obtain information about the

senses from lexical resources. They are less restricted in terms of data, while offering wider coverage. They still suffer from several problems, however, largely as a result of the fact that the lexical resources are not anchored in the target domain. A detailed examination of these issues, along with examples, is presented in Section 4.3.2.

Our own work uses insights gained from unsupervised methods with the aim of creating large datasets of sense-labeled instances without explicit manual coding. Similarly to McCarthy et al. (2004), we assume that words related to the target word are useful indicators of its senses. Unlike McCarthy et al. (2004), however, our method disambiguates words in context and is able to assign additional senses, besides the first one. Our approach leverages the information provided by a lexical resource, but unlike the lexical methods mentioned previously, is grounded in the domain of interest through our use of distributional similarity. It does not require parallel corpora, and is free of the restrictions of the translation-based approaches.

4.3 Methodology

4.3.1 Overview

We start off with the observation that different senses of a word tend to occur with different contextual features, which can be used for disambiguation. This assumption is the basis for most word-sense disambiguation algorithms, both supervised and unsupervised, and dates back to Weaver (1949/1955). As an example, if we were given the ambiguous word *bat*, and told that the playing-stick sense of *bat* tends to occur near the word *ball*, whereas the animal sense tends to be the subject of the verb *fly*, we can use these contextual features to disambiguate instances where one of these features is present.

Standard supervised approaches to WSD use a variety of machine learning methods to learn such contextual sense-cues from a large training set containing many examples of the target ambiguous word, each provided with a portion of local context and annotated with the correct sense. In order to alleviate the need for manually annotated data, the approach proposed here makes use of some of the principles first presented in McCarthy et al. (2004). They made use of a combination of distributional and semantic similarity measures in order to infer the predominant sense of a word in the corpus in an unsupervised fashion (see Section 3.2.3.2). In our work, we take this approach several steps forward, and use the combination of similarity methods in order to produce

```
For each ambiguous word:

1. acquire neighbors of the target word (i.e., words which are
   likely to share context features with certain senses of the
   target).

2. associate each neighbor with the relevant sense(s).

3. replace instances of neighbors in the corpus with
   sense-labelled instances of the target word.

4. train a supervised classifier for senses using the generated
   data.
```

Figure 4.1: Outline of method for producing pseudo-labeled training data

a sense-labeled training set, without resorting to human intervention. This dataset can be used by any supervised classifier to preform WSD.

Figure 4.3.1 summarizes our method. Similarly to McCarthy et al. (2004), we first use a distributional similarity measure to obtain a list of words (distributional neighbors) similar to the word we wish to disambiguate. The distributional information is gathered from a large corpus of (un-annotated) text, in the domain of interest. After acquiring the neighbors, our method diverges from that of McCarthy et al. (2004). Whereas they use a semantic similarity metric to score the senses of the target word and select the predominant one, we employ the semantic measure to separate the neighbors into sense-specific groups. Once each neighbor is associated with a target sense, we proceed with the creation of labeled data. We extract occurrences of each of the neighbors in our corpus, and transform them into an instance of the target word, labeled with the matching sense for that neighbor. The important steps, from an unsupervised perspective are: (1) acquiring neighbors, and (2) associating neighbors with senses. Each step can be implemented in several ways. We describe in more detail our choice of implementation for each of these stages below.

Despite the simplicity of the method, we demonstrate that the resulting labelled

training set can be successfully used for training supervised classifiers for WSD, which outperform other state-of-the-art unsupervised methods and are only slightly inferior to the performance of the same classifiers using expensive manually-annotated data.

Neighbor Acquisition There are many means of obtaining appropriate neighbors for the target word. Broadly speaking, these fall into two categories: the neighbors can be extracted from a corpus (distributional neighbors) or from a semantic resource, for example the dictionary providing the sense inventory (semantic neighbors). A wealth of algorithms have been proposed in the literature for acquiring distributional neighbors from a corpus (see Weeds 2003 for an overview). They differ as to which features they consider and how they use the distributional statistics to calculate similarity.

Lin's (1998a) information-theoretic similarity measure is commonly used in lexicon acquisition tasks and has demonstrated good performance in unsupervised WSD (McCarthy et al., 2004). It operates over dependency relations pertaining to the target word. For example, in the sentence *"The big bat flew into the cave"*, the word *bat* participates in two dependency relations – it is the subject of the verb *flew*, and is modified by the adjective *big*.

In Lin's (1998a) similarity measure, a word w is described by a set $T(w)$ of co-occurrence triplets $< w, r, w' >$, where r represents the type of relation (e.g., *object-of*, *subject-of*, *modified-by*) between w and its dependent w'. A word's triplet set can be viewed as a sparsely represented feature vector for that word. The similarity between w_1 and w_2 is then defined as:

$$S(w_1, w_2) = \frac{\sum_{(r,w) \in T(w_1) \cap T(w_2)} I(w_1, r, w) + I(w_2, r, w)}{\sum_{(r,w) \in T(w_1)} I(w_1, r, w) + \sum_{(r,w) \in T(w_2)} I(w_2, r, w)} \tag{4.1}$$

where $I(w, r, w')$ is the *information value* of w with regard to (r, w'), defined as:

$$I(w, r, w') = \log \frac{count(w, r, w') \cdot count(r)}{count(*, r, w') \cdot count(w, r, *)} \tag{4.2}$$

The measure is used to estimate the pairwise similarity between the target word and all other words in the corpus (with the same part of speech); the k words most similar to the target are selected as its neighbors.

A potential caveat with Lin's (1998a) distributional similarity measure is its reliance on syntactic information in the form of dependency relations. An accurate dependency parser may not be available for all languages or domains, thereby restricting

the use of this measure. An alternative is to use a measure of distributional similarity which does not use syntactic information. One such measure is InfoMap[1], which considers only word co-occurrence statistics and therefore does not require a syntactic parser. In InfoMap, each word is represented by a vector which records how many times it co-occurs with other words. The similarity between any two words can then be measured using a vector-distance measure such as cosine. Syntax-free distributional neighbors have been previously used in word sense discrimination and disambiguation (Schütze, 1998; Dagan et al., 1997). Syntax-free measures, such as InfoMap, have the advantage of usability in cases where accurate dependency parsers are unavailable. However, dependency information is often less noisy, since it pertains directly to the word of interest, whereas co-occurring words may be not be directly related. This makes dependency-based measures more accurate, and a preferred choice when syntactic information can be obtained.

As mentioned earlier, it is also possible to obtain neighbors simply by consulting a semantic dictionary. In WordNet, for example, we can assume that WordNet relations, (e.g., hypernymy, hyponymy, synonymy) indicate words which are semantic neighbors. An advantage of using distributional neighbors is that they reflect the characteristics of the corpus we wish to disambiguate and are potentially better suited for capturing sense differences across genres and domains, whereas dictionary-based neighbors ignore the corpus and domain. These issues, and their effects on WSD performance, are further discussed in Section 4.6.

Associating Neighbors with Senses If the neighbors are extracted from WordNet, it is not necessary to associate them with their senses as they are already assigned a specific sense. Distributional similarity methods, however, do not provide a way to distinguish which neighbors pertain to each sense of the target. For that purpose, we adapt a method proposed by McCarthy et al. (2004). Specifically, for each acquired neighbor, we choose the sense of the target which gives the highest semantic similarity score to *any* sense of the neighbor. There are a large number of semantic similarity measures to choose from (see Budanitsky and Hirst 2001 for an overview). We use Lesk's measure as modified by Banerjee and Pedersen (2003) for two reasons. First, it has been shown to perform well in the related task of predominant sense detection (McCarthy et al., 2004). Second, it has the advantage of relying only upon the sense definitions, rather than the complex graph structure which is unique to WordNet. This

[1] `http://infomap.stanford.edu`

makes the method more suitable for use with other sense inventories.

Note that unlike McCarthy et al. (2004), we are associating neighbors with senses, rather than merely trying to detect the predominant sense, and therefore require more precision in our selection. When it is unclear which sense of the target word is most similar to a given neighbor (i.e., when the scores of two or more senses are close together), that neighbor is discarded. The degree of 'closeness' is a parameter which depends on the data at hand and is tuned experimentally (in our experiment, we discarded neighbors where the two most similar senses were within 20% of each other).

4.3.2 Example

As an example, we will show the complete process for disambiguating a word, using the two neighbor types described above: distributional neighbors, using a large corpus (BNC), and semantic neighbors from a lexical resource (WordNet). This choice of corpus and lexical resource are natural ones, since our test data (from Senseval 2) is composed of instances from the BNC, and uses WordNet as the sense inventory. The word *sense* has five senses in WordNet, of which two were grouped together by the Senseval 2 annotators, to form the following four coarse-grained senses.

1. a. A general conscious **awareness**.
   (e.g., *a sense of security*)

   b. The faculty through which the external world is apprehended.
   (e.g., *a sense of smell*)

2. The **meaning** of a word.
   (e.g., *The dictionary gave several senses for the word*)

3. Sound practical **judgment**.
   (e.g., *I can't see the sense in doing it now*)

4. A natural appreciation or **ability**.
   (e.g., *keen musical sense*).

The first stage in the process involves the acquisition of sense-specific neighbors. Using the Lin (1998a) distributional method described above on the BNC corpus, the following neighbors were retrieved and associated with the relevant senses:

- **Neighbors of sense 1:** awareness, feeling, instinct, enthusiasm, sensation, vision, tradition, consciousness, anger, panic, loyalty
- **Neighbors of sense 2:** emotion, belief, meaning, manner, necessity, tension, motivation

No neighbors are associated with the last two senses, indicating that they are not prevalent enough in the corpus to be detected through the distributional similarity method. This is borne-out by the sense frequencies in the test data, where the first two senses comprise 38% and 39% of the instances respectively, and the third and fourth senses account for only 19% and 4%.

Using semantic neighbors from WordNet, as described above, results in the following sense-specific neighbors (excluding words not present in our corpus):

- **Neighbors of sense 1:** sentience, sensation, sensitivity, sensitiveness, sensibility, modality, module, knowingness, faculty, consciousness, cognizance, cognisance, awareness, will, volition, understanding, speech, self-awareness, retentiveness, retention, reason, memory, language, intellect, feel, attention
- **Neighbors of sense 2:** signified, acceptation, signification, significance, meaning, import, symbolization, symbolisation, subtlety, spirit, shade, refinement, referent, purport, point, overtone, nuance, nicety, moral, lesson, intent, intention, gist, essence, effect, core, connotation, burden
- **Neighbors of sense 3:** gumption, logic, sagacity, judgment, judgement, discernment, prudence, judiciousness, eye, discretion, circumspection
- **Neighbors of sense 4:** hold, grasp, appreciation

Wordnet, of course, contains information about all the senses it lists. We see, however, that rarer, more specific, senses have fewer neighbors in WordNet (which are attested in our corpus). We can also see one of the main problems with acquiring neighbors from a semantic resource. Many of the neighbors are themselves ambiguous, and only one of their senses (often a rare one) is related to our target sense. This is the case, for example, with the words *modality* and *faculty* associated with sense 1, the word *shade* associated with sense 2, and many more. Such neighbors may provide noisy and misleading information.

A common approach to rectifying this problem is to select only monosemous related words. However, this approach tends to have the unwanted effect of removing

many informative neighbors along with the noisy ones. These are the monosemous neighbors from the previous list:

- **Neighbors of sense 1:** cognisance, self-awareness
- **Neighbors of sense 2:** signified, signification, nuance, moral, intention

All neighbors of the two rarer senses were eliminated, since they are ambiguous. Unlike the distributional neighbors, where the absence of neighbors for the last two senses was an indication of their rarity, here this represents a limitation of the resource and method used. However, the neighbors that remain are arguably less noisy and more specific to the associated sense.

The advantages and disadvantages of both neighbor-acquisition methods can be seen in the example above. While WordNet provides neighbors for all senses, they are not necessarily relevant to the specific corpus, and do not exhibit the "replaceability" that the distributional neighbors do. In other words, they may have similar meanings, but are not usually used in the same type of context, so are less useful for extracting helpful contextual cues. The WordNet hierarchy is abstract, and often the categories don't fit the common usage or distinctions made in the text. In addition, WordNet is primarily a dictionary, not a thesaurus, and so lists only words which are strongly synonymous as synonyms. Words that are related less directly (e.g., hypernyms, hyponyms) vary in their degree of synonymy, and are sometimes semantically quite different, which again poses an obstacle when trying to learn contextual cues.

Once sense-specific neighbors are acquired, by one of the methods described above, the next stage is to replace instances of the neighbors in our corpus with the target ambiguous word labeled with the appropriate sense. For example, when encountering the sentence *"The philosophical explanation of authority is not an attempt to state the* ***meaning*** *of a word"* in the corpus, our method would automatically transform this to *"The philosophical explanation of authority is not an attempt to state the* ***sense*** *(s#2) of a word."* This is done for every sentence in the corpus containing a neighbor. These modified sentences, or *pseudo-instances*, comprise the training corpus we provide to each of our machine learning algorithms. The classifier trained on these instances is then used for disambiguating genuine instances of the ambiguous target word. Note that if we were using only monosemous semantic neighbors, which do not include the neighbor *meaning*, our method would ignore this sentence in the corpus, and we would have less training examples. This means that different lists of neighbors lead to

completely different training datasets, which may also vary considerable in size (see Table 4.2)

4.4 Experimental Setup

4.4.1 Test Data

For the purpose of our experiments, we made use of the data provided for the English lexical sample task in Senseval 2 (Preiss and Yarowsky, 2001) and Senseval 3 (Mihalcea and Edmonds, 2004) workshops. Since our methods are unsupervised, and therefore do not make use of the labels in the training data, we were able to merge the training and test data for use in our evaluations.

Since our method does not require sense-tagged data, it could be applied to the disambiguation of any word in the lexicon. This means it could be used for disambiguation in the all-words task. However, our approach is not strictly an online method, since the procedure requires the unsupervised construction of a separate training set for each word we wish to disambiguate, and training a machine-learning classifier for that word using the dataset, before proceeding with the actual disambiguation. We chose the lexical sample task as a proof of concept, to demonstrate that the method is feasible and successful.

In our experiments, we made use of the coarse-grained sense grouping provided for both Senseval datasets. It is widely recognized (see Edmonds and Kilgarriff 2002, Navigli 2006, Snow et al. 2007) that differing levels of granularity are suitable for different tasks. For many NLP applications, coarse grained differences are more suitable (see, for example, Moldovan and Mihalcea 2000), and finer distinctions may cause more harm than good. In order to assess the effect of granularity on our method, we also performed an experiment comparing the results on coarse- and fine-grained sense distinctions.

The workshop organizers provided a small amount of surrounding context for each instance (usually a sentence or two surrounding the sentence containing the target word). This context was parsed using RASP (Briscoe and Carroll, 2002), to extract part-of-speech tags, lemmatized forms of the words, and dependency information, from which we extracted the feature representation of our instances (see Section 4.4.3). We filtered the data by removing all instances for which the annotators disagreed on the correct tagging. We also removed instances which were not correctly recognized

by the parser (a target word tagged with the wrong part-of-speech, for example). These comprised only 1.6% for the Senseval 3 data, but almost 18% of the Senseval 2 data. The reason for this is the large number of multi-word expressions in Senseval 2, and the inclusion of words that are more commonly verbs than nouns (such as *grip*), which confused the parser. This was done to isolate the results of our system from the effects of external processes, such as the accuracy of the parser. In cases where more than one instance of the target word existed in the provided context, we disambiguated the first mention, in order to eliminate the problems of identifying the correct instance of the word. The original data consisted of 4835 training and test instances for Senseval 2, and 4997 for Senseval 3. After filtering we retained 2985 instances for Senseval 2, and 4652 for Senseval 3.

As can be observed in Table 4.1, the two Senseval datasets differ considerably. The Senseval 3 data has a higher level of ambiguity, and is therefore a more difficult dataset. In addition, although Senseval 3 has a slightly lower percentage of first sense instances, the higher ambiguity means that the skew is, in fact, much greater than in Senseval 2. A large skew towards the predominant sense means there are less instances from which we can learn about the rarer senses, and that we run a higher risk when labeling an instance as one of the rarer senses (instead of defaulting to the predominant one).

If we had access to an oracle, and labeled each word in our test data with its true predominant sense, we would achieve 66.96% accuracy on the Senseval 2 dataset, and 62.15% accuracy on Senseval 3.

4.4.2 Automatically Created Training Data

As mentioned in Section 4.3 we retrieved neighbors using Lin's (1998a) similarity measure on a RASP parsed (Briscoe and Carroll, 2002) version of the BNC. We used subject and object dependencies, as well as adjective and noun modifier dependencies. We also created training data sets using collocational neighbors. Specifically, using the InfoMap toolkit[2], we constructed vector-based representations for individual words from the BNC using a term-document matrix and the cosine similarity measure. Vectors were initially constructed with 1,000 dimensions, the most frequent content words. The space was reduced to 100 dimensions with singular value decomposition (Berry et al., 1994). From the neighbors returned by the system, we selected only those

[2] `http://infomap.stanford.edu/`

Senseval 2	# tags	amb.	1st sense
art	84	2	48 (57%)
authority	113	4	60 (53%)
bar	267	7	176 (65%)
bum	94	2	80 (85%)
chair	184	2	163 (88%)
channel	57	4	29 (50%)
child	165	2	104 (63%)
church	145	3	76 (52%)
circuit	97	4	44 (45%)
day	331	5	206 (62%)
dyke	47	2	35 (74%)
facility	148	2	146 (98%)
fatigue	61	3	54 (88%)
feeling	132	4	81 (61%)
grip	31	5	21 (67%)
hearth	51	2	45 (88%)
material	155	3	81 (52%)
mouth	128	3	113 (88%)
nation	75	2	60 (80%)
nature	116	3	72 (62%)
post	148	4	86 (58%)
restraint	63	4	29 (46%)
sense	122	4	51 (41%)
spade	80	3	60 (75%)
stress	91	3	70 (76%)
total	2,985	3.28	66.96%

Senseval 3	# tags	amb.	1st sense
argument	259	4	149 (57%)
arm	387	5	308 (79%)
atmosphere	173	5	112 (64%)
audience	259	2	244 (94%)
bank	371	8	278 (74%)
degree	340	5	231 (67%)
difference	297	4	179 (60%)
difficulty	64	2	59 (92%)
disc	262	4	110 (41%)
image	208	6	86 (41%)
interest	270	6	116 (42%)
judgment	94	5	29 (30%)
organization	164	4	136 (82%)
paper	206	4	104 (50%)
party	309	4	216 (69%)
performance	234	3	105 (44%)
plan	207	2	157 (75%)
shelter	240	4	113 (47%)
sort	226	3	184 (81%)
source	82	7	45 (54%)
total	4,652	4.35	62.15%

Table 4.1: Properties of the Senseval 2 and 3 lexical sample datasets used as test data. For each word, we give the total number of labeled examples we used (#tags), the ambiguity, or number of (coarse-grained) senses (amb.), and the number (and percentage) of instances labeled with the most frequent sense (1st sense).

Dataset	Depend	Co-Occur	AllWN	MonoWN
Senseval 2	172K	110K	246K	12.8K
Senseval 3	168K	135K	412K	14.5K

Table 4.2: Number of training instances obtained with our method when using dependency-based (Depend) and co-occurrence based (Co-Occur) distributional neighbors, unfiltered WordNet neighbors (AllWN), and monosomous WordNet neighbors (MonoWN).

which were nouns. Since the algorithm relies only on co-occurrence statistics, it returns relevant words of any part-of-speech. However, our experiments focus solely on nouns. Furthermore, using neighbors from other parts-of-speech as pseudo-instances of our target ambiguous noun would introduce noise, and be likely to confuse the machine learning classifiers.

Finally, we also extracted neighbors from WordNet by selecting synonyms, antonyms, hyponyms, hypernyms and siblings (i.e., hyponyms of the same hypernym) of the target word, in that order. A problem often encountered when using dictionary-based neighbors is that they are themselves polysemous, and the related sense is often not the most prominent one in the corpus, which leads to noisy data. We therefore experimented with using *all* neighbors for a given word or only those which are *monosemous* and hopefully less noisy. In all cases we used 50 neighbors, the most similar nouns to the target. Table 4.2 shows the number of training data instances we obtained according to the different neighbor selection methods.

4.4.3 Feature Space

In order to represent the training and test instances in our supervised learning setup, we used a feature set designed to capture both immediate local context, wider context and syntactic context. We used six feature categories: $\pm$10-word window, $\pm$5-word window, collocations, word n-grams, part-of-speech n-grams and dependency relations (including verb-object, verb-subject, adjective-noun modifiers, and noun-noun modifiers). These feature types have been widely used in various WSD algorithms (see Lee and Ng 2002 for a detailed evaluation). An example instance with its feature vector (containing features from all the categories we used) is presented in Figure 4.2. These feature types have been widely used in various WSD algorithms (see, for instance, Florian et al. 2002 and see Lee and Ng 2002 for an evaluation of the effectiveness of

each of these feature categories to WSD). Our feature space is of high dimensionality, containing every possible feature in the categories listed, but is very sparse, since only a small number actually occur in the data. In all cases, we use the lemmatized version of the word(s).

4.4.4 Supervised Classifiers

We experimented with three supervised classifiers, which are based on different learning paradigms. This allows us to examine the effect of our training-data creation procedure on different kinds of classifiers in order to judge which are most suited for use with our method. All these classifiers have been previously used for the purpose of WSD, and have shown competitive performance (see Niu et al. 2005b, Preiss and Yarowsky 2001 and Mihalcea and Edmonds 2004).

Support Vector Machine (SVM) SVMs model classification as the problem of finding a separating hyperplane in a high dimensional vector space. SVM classifiers focus on differentiating between the most problematic cases - instances which are close to one another in the high dimensional feature space, but have different labels. The SVM classifier is discriminative, rather than generative, and does not explicitly model the classes. SVMs have been used very successfully in many NLP tasks. We used the multi-class bound-constrained support vector classification (SVC) version of SVM described in Hsu and Lin (2001) and implemented in the BSVM package[3]. The only parameter we provided was the misclassification penalty. We set this to a high value (1000), in order to avoid labeling all instances with the most frequent sense.

Maximum Entropy Maximum Entropy based classifiers are a common alternative to other probabilistic classifiers, such as Baysian classifiers, and have received much interest in various NLP tasks, such as part-of-speech tagging (Ratnaparkhi, 1996)and text classification (Nigam et al., 1999). Maximum Entropy classifiers represent a probabilistic, model-based, global constrained approach. They assume a uniform, zero-knowledge (maximal entropy) model, under the constraints of the training dataset. The classifier finds the (unique) maximal-entropy model which comforms to the expected feature distribution of the training data. Maximum Entropy classifiers tend to overfit when provided with only small amounts of training data. In our case, where there is an

[3] http://www.csie.ntu.edu.tw/~cjlin/bsvm/

*"The philosophical explanation of authority is not an attempt to state the **sense** of a word."*

Topical Features	
General Context (ten preceding and ten following words)	explanation of authority is not ...
Local Context (five preceding and five following words)	an attempt to state ...
Collocation Features	
Preceding Word Next Word	the of
Word N-Grams	
Preceding Bigram Flanking Bigram Following Bigram	state the X the X of X of a
Part-of-Speech N-Grams	
Preceding Bigram Flanking Bigram Following Bigram	Verb Det. X Adj. X Prep. X Prep. Det.
Dependencies	
Verb-Object Dependency	object of verb state

Figure 4.2: Example sentence representing an instance of the target word *sense*. The table lists the feature types (left) and their associated values (right).

abundance of (automatically) labeled data, we are more concerned with how well the classifier handles the noise introduced by our automatic labeling scheme. We used the optimized Megam implementation (Daumé III, 2004)[4].

Label Propagation The basic Label Propagation algorithm (Zhu and Ghahramani, 2002) represents labeled and unlabeled instances as nodes in an undirected graph with weighted edges. Initially only the known data nodes are labeled. The goal is to propagate labels from labeled to unlabeled points along the weighted edges. The weights are based on distance in a high-dimensional space. At each iteration, only the original labels are fixed, whereas the propagated labels are "soft", and may change in subsequent iterations. This property allows the final labeling to be affected by more distant labels, that have propagated further, and gives the algorithm a global aspect. We used SemiL[5], a publicly available implementation of the label propagation algorithm (and set all the parameters to the default values).

The purpose of this work is not to determine the best settings and parameters for each of these classifiers, but rather to determine which classifier(s) work best with our method of label generation, and how each is affected by use of labeled pseudo-instances, rather than real human-annotated ones. For this reason, we made no attempt to optimize the parameters of any of these classifiers, and tried to use the basic, out-of-the-box settings. These settings are used for all our experiments. In addition, we use the same feature space, throughout. This is a simple agglomeration of features commonly used for WSD, as described in Section 4.4.3. We did not attempt to manually optimize the feature space to the task in any way.

4.4.5 Baselines and Comparisons

As an upper bound on expected accuracy, we compare to the results of the same classifiers when using manually-labeled data (under the same experimental settings). This provides an estimate of the expected decrease in accuracy caused solely by the use of our automatic data-labelling method. Given a more successful classifier, or a better set of parameters for this task, which increases the accuracy on hand-labelled data, we can expect a similar increase when using our automatically-labelled data.

[4] `http://www.cs.utah.edu/~hal/megam/`

[5] `http://www.engineers.auckland.ac.nz/~vkec001`

We also compare our method to two other unsupervised ones. The first of these is the Lesk algorithm, which is completely local, and uses only immediate context and dictionary definitions for token disambiguation. The second is the unsupervised predominant-sense algorithm of McCarthy et al. (2004), which is type-based and global, and ignores local context. A more detailed description of both these algorithms can be found in Section 3.2. We used an augmented version of the Lesk algorithm, where unknown instances (where no overlap was found) are tagged using an automatically-derived predominant sense (via the method of McCarthy et al. 2004).

Throughout our experiments, we use the χ^2 test to determine the statistical significance of performance differences. When stating that the results of two systems are significantly different, we mean $p < 0.01$, unless otherwise noted.

4.5 Results

4.5.1 System Performance

Table 4.3 presents the results of the various algorithms trained on automatically generated (four center columns) and the manually tagged data (rightmost column). We report the percentage of correctly labeled instances (since all algorithms labeled all instances, accuracy, precision, recall and F-score are all equivalent).[6]

Manually Labeled Data In order to obtain an upper bound and measure of comparison for our experiments, we performed a 5-fold cross-validation test using the human annotation provided with the Senseval dataset. For this purpose, we randomly divided the Senseval data (the combined testing and training, as mentioned in Section 4.4.1) into five portions. We then performed five experiments, in each one we used a single 20% portion as test data, and the remaining 80% for training. We ran each of our supervised classifiers on the data, and averaged the results of the five experiments. The rightmost column of the table shows the resulting average accuracy scores. As previously stated (Section 4.4.4), these scores do not represent the highest achievable performance in a supervised setting, but rather those obtained using the various algorithms in their most basic, 'out-of-the-box' setting.

[6] As mentioned in the previous section, the results in the table represent an augmented version of the Lesk algorithm. For the unaugmented Lesk algorithm, the results are: For Senseval 2, F-score = 36.93% ($precision = 41.78\%, recall = 33.10\%$). For Senseval 3, F-score = 37.00% ($precision = 41.67\%, recall = 33.28\%$)

Senseval 2	AllWN	MonoWN	Co-Occur	Depend	Manual
SVM	49.45%	51.62%	60.57%	63.72%	72.63%
MaxEnt	41.34%	49.28%	60.23%	61.91%	72.70%
LP	41.27%	47.91%	59.20%	63.42%	69.65%
McCarthy	60.64%				
Lesk	45.43%				

Senseval 3	AllWN	MonoWN	Co-Occur	Depend	Manual
SVM	51.46%	44.73%	55.98%	58.60%	74.91%
MaxEnt	47.89%	43.10%	54.88%	57.93%	74.83%
LP	45.74%	43.36%	62.06%	62.06%	69.88%
McCarthy	56.58%				
Lesk	43.98%				

Table 4.3: Accuracy on Senseval 2 and 3 lexical samples. Support vector machine (SVM), maximum entropy (MaxEnt) and label propagation (LP) classifiers are trained on automatically and manually labeled data sets.

As we can see, all the algorithms exhibit a similar level of performance, with slightly lower scores for the Senseval 3 dataset, which is more difficult, as mentioned in Section 4.4.1. On the Senseval 2 data, the SVM is significantly better that the other two classifiers. For Senseval 3, Label Propagation is significantly worse than the others. All other differences are not significant. The results shown here are comparable to those achieved by the state-of-the-art supervised systems participating in the two Senseval competitions. The best performing systems in the Senseval competitions (on coarse-grained nouns) achieved approximately 76% accuracy in Senseval 2, and close to 80% accuracy in Senseval 3. The SMU system (Mihalcea, 2002) achieved the best results on the Senseval 2 English lexical sample task. It used a seed set of noun phrases and verb-noun constructs created from existing sense-annotated data (WordNet and SemCor), and from web queries on closely related monosemous words from WordNet. The seed set was expanded by using web queries on the elements in the original set and extracting other phrases containing the target word from the retrieved contexts. Patterns of local context (up to two words surrounding the target) were learned from the expanded example set using a set of heuristics. The training set provided by the task organizers was used to filter and remove erroneous patterns. The patterns were

used for disambiguation when present in the context of the test instances. In cases where there is no contextual information, the first sense from WordNet was used.The Basque Country University system (Agirre and Martínez, 2004) was the best performing system in the Senseval 3 English lexical sample task. It combined the output of four learning algorithms (Decision Lists, Naive Bayes, a vector space model and a Support Vector Machine). The system used a rich feature space which included syntactic dependencies and domain information extracted using different tools and also from external resources. Different smoothing methods were tested on Senseval 2 data, and the best parameters for the systems and their combination were calculated using 10-fold cross-validation on the Senseval 3 training data. It is likely that the higher scores achieved by the state-of-the-art systems in Senseval 3 are due to better systems and algorithms, rather than a decrease in task difficulty (as explained in Section 4.4.1, the Senseval 3 task is more difficult in several respects). For instance, many of the systems in Senseval 3, including the best performing one, were combinations of different classifiers (see Mihalcea and Edmonds 2004). Also, the number of training instances per word was considerably larger in Senseval 3 (see Section 4.4.1).

Automatically Labeled Data The results using the automatically acquired training sets are presented in the four center columns. Each column represents one source of neighbors used to create the pseudo-labeled data. Using the neighbors provided by WordNet (AllWN) leads to significantly lower scores than the use of either co-occurence-based (Co-Occur) or dependency-based (Depend) distributional neighbors. Using only monosemous neighbors from WordNet (MonoWN) improves the results slightly on the Senseval 2 dataset, though they are still significantly lower than those resulting from the distributional neighbors. On the Senseval 3 dataset, using only monosemous neighbors lowers the accuracy. A possible explanation for this is that the WordNet neighbors for Senseval 2 are very noisy, so that filtering out polysemous neighbors helps improve the accuracy. For the Senseval 3 words, the noise is less, so that the expected gain from reduction of noise is out-weighed by the amount of information lost in the filtering (see Section 4.3.2 for an example). Another reason may lie with the choice of words in each of the lexical samples. In the Senseval 2 dataset, the main criterion for selection was to give a range of low, high and medium frequency words in the chosen corpus (BNC, Kilgarriff 2001). No such criterion was given for the choice of lexical sample words in Senseval 3. The higher average ambiguity (in coarse grained senses) also suggests higher frequency. This may lead to differences in

the type and ambiguity of the neighbors retrieved.

When comparing the results of using co-occurrence-based distributional neighbors (Co-Occur) to using dependency-based ones (Depend), we can see a drop of 1.5-4.5% accuracy on the Senseval 2 test set, and less than 3% on the Senseval 3 test set. As mentioned in Section 4.3.1, the co-occurrence-based distributional neighbors can be acquired even when parsing tools are unavailable. Our experiments demonstrate that using distributional neighbors is still a viable option in such cases, and achieves results which are almost as good as when dependency information is available.

The results using both types of distributional neighbors are all significantly better than those using neighbors (monosemous or otherwise) from WordNet, regardless of the classifier used. We discuss the reasons for this in Section 4.6. On both datasets, dependency-based distributional neighbors perform better than using the automatically-acquired predominant sense (for the Maximum Entropy classifier, the difference is not statistically significant).

When we compare the results from the manually tagged data to those achieved by using the distributional neighbors, we can see that use of our pseudo-tagged data results in scores that are approximately 7-10% lower for Senseval 2, and 8-17% lower for Senseval 3. Since the results were achieved using the same feature set and classifier settings, the comparison provides an estimate of the expected decrease in accuracy due only to our unsupervised tagging method. The implication is that for a given supervised method, we can expect, using our (automatically constructed) dataset, to do approximately 90% as well as we would if we had a manually tagged dataset the size of the one used in the Senseval 2 experiments, or 85% as well as with one the size of the Senseval 3 training set. Our method allows for any improvement in supervised WSD algorithms to be easily transferred to unsupervised WSD, by using our automatically-constructed training dataset instead of a manually-annotated one.

4.5.2 Coverage

Table 4.4 shows the percentage of instances labeled with senses other than the most frequent, in each of the experimental settings. If we compare to Table 4.3 we can see that these numbers are inversely correlated with accuracy. As stated in Section 4.4.1, the decision to label with a secondary sense is a risk, since the first sense is usually very dominant. On the other hand, we are not interested in labeling all instances with a single sense, since this defeats the purpose of context-dependent WSD. Table 4.5

Senseval 2	AllWN	MonoWN	Co-Occur	Depend	Manual
SVM	31.26%	25.23%	10.99%	12.50%	19.70%
MaxEnt	15.98%	16.28%	3.45%	5.49%	18.83%
LP	23.08%	19.77%	3.62%	8.91%	13.87%

Senseval 3	AllWN	MonoWN	Co-Occur	Depend	Manual
SVM	38.13%	28.98%	12.94%	18.10%	24.10%
MaxEnt	31.23%	22.46%	6.26%	8.71%	23.52%
LP	30.95%	12.36%	1.74%	5.37%	16.60%

Table 4.4: Percentage of instances labeled with secondary senses when using automatically and manually labeled training data.

Senseval 2	AllWN	MonoWN	Co-Occur	Depend	Manual
SVM	43.41%	36.79%	53.05%	58.71%	62.59%
MaxEnt	48.43%	50.41%	73.79%	65.24%	64.23%
LP	40.49%	39.83%	57.41%	65.04%	58.70%

Senseval 3	AllWN	MonoWN	Co-Occur	Depend	Manual
SVM	37.54%	48.44%	41.36%	44.77%	65.57%
MaxEnt	33.59%	40.67%	40.89%	60.25%	65.90%
LP	31.81%	53.91%	66.67%	58.00%	59.72%

Table 4.5: Classifier accuracy for secondary sense labels when using automatically and manually labeled training data.

	Coarse-Grained		Fine-Grained	
Senseval 2	Depend	Manual	Depend	Manual
SVM	63.72%	73.53%	47.00%	64.79%
MaxEnt	61.91%	73.53%	43.05%	65.13%
LP	63.42%	69.65%	40.57%	43.35%

	Coarse-Grained		Fine-Grained	
Senseval 3	Depend	Manual	Depend	Manual
SVM	58.60%	76.85%	53.44%	69.05%
MaxEnt	57.93%	76.20%	53.10%	68.96%
LP	62.06%	69.88%	46.17%	42.52%

Table 4.6: Comparison of classifier performance on fine- and coarse-grained sense distinctions.

gives the accuracy of the classifiers with regard to only the secondary senses. Here we see further evidence that the choice to label more instances with a non-first sense is risky, leading to lower accuracy. As expected, this effect seems to be stronger in the Senseval 3 data, for the reasons mentioned in Section 4.4.1.

It is interesting to note that the SVM classifier labels two to three times as many instances with secondary-sense labels, while still achieving similar levels of overall accuracy to the other classifiers (Table 4.3) and only slightly lower accuracy on the secondary senses (Table 4.5). This fact would make it a better choice when it is important to have more data on rarer senses. Another point of interest is the fact that even the SVM classifier strongly under-represents the rarer senses. This is especially true when using distributional neighbors. Since the few most frequent senses account for almost all of the occurrences of the target word in the corpus, and therefore comprise almost all its distributional probability mass, distributional-similarity metrics rarely provide neighbors for more than the two or three most frequently occurring senses.

4.5.3 Fine-Grained Senses

In our initial experiments, we decided to evaluate our method on coarse-grained sense distinctions, for the reasons stated in Section 4.4.1. In order to determine the impact of this decision on our results, we preformed a comparison experiment using dependency-

Method	F-Score
SVM	53.44%
MaxEnt	53.10%
LP	46.17%
Rank	48.41%
ProbMix	46.17%

Table 4.7: Comparison of performance between classifiers trained using dependency-based automatically-labeled data and rank-based (Rank) and probability-mixture (ProbMix) ensembles on Senseval 3 fine-grained senses.

based distributional neighbors on fine-grained sense distinctions. Table 4.6 shows the scores for this experiment. When moving from coarse to fine-grained senses, there is a significant difference between the two Senseval datasets. The average ambiguity in Senseval 2 increases from 3.28 senses to 5.6 senses, whereas in Senseval 3 the increase is much smaller – from 4.35 senses to 4.8 senses on average. The greater increase in ambiguity for the Senseval 2 dataset leads to a correspondingly large decrease in accuracy, when compared to Senseval 3. However, in both cases, we see that the classifiers' sensitivity to the granularity shift is similar when trained on our automatically-labeled data (Depend) and on the manually-labeled data (Manual). This indicates that our data creation method is not particularly sensitive to the granularity, and is applicable for a variety of levels of sense distinction.

4.5.4 Comparison to Ensemble Methods

In Chapter 3 we presented several ensembles which combine the output of a group of WSD algorithms in order improve performance. The ensembles make use of the predominant-sense type-based approach. In other words, they estimate the most frequent sense of each word in the data, and label all instances of the word with that sense. In Chapter 3 we evaluated the ensembles in an all-words setting, where the large number of words allowed an accurate assessment of their success in estimating the true predominant sense. For the sake of completeness, we compare our context-specific data-creation method and the top ensembles from Chapter 3 on a lexical sample task. Table 4.7 presents the results of the classifiers trained using our dependency-based distributional neighbors method, and the rank-based (Rank) and probability mixture (ProbMix) ensembles, on Senseval 3 fine-grained senses. Both the SVM and the

maximum-entrophy classifiers do significantly better than the ensembles. The label-propogation classifier, which has the weakest performance on this data, does as well as the probability mixture ensemble, but worse than the rank-based one. However, despite its relatively weak performance, it still has the advantage of performing context specific WSD, and labeling some instances with secondary senses.

4.6 Discussion

Our experiments addressed and provided information about several important issues regarding WSD through the use of automatically created training data. The first choice one has to make is the method of data creation. As mentioned in Section 4.2.2, the translation-based approach is restricted by the relative scarcity of parallel corpora and limited to words and senses which are distinguished by different translations. Methods based on semantic neighbors from a lexical resource seem less constrained and more widely applicable. However, our experiments clearly show that resource-based neighbors are less effective for creation of labeled data than distributional ones. There are several reasons for this:

Topical vs. Local Information one characteristic of the lexical-resource approach is that while the related words share meaning with the target, they often do not share local behavior. In other words, they do not appear in the same immediate local context, do not share syntax, or are used differently in the sentence. For this reason, the useful information that can be extracted from their contexts tends to be topical (e.g., informative words in the document which are indicative of a general topic or domain), rather than local (e.g., part-of-speech of words which are adjacent to the target, grammatical dependencies etc.). According to Leacock et al. (1998), topical features appear to be more useful for the disambiguation of nouns, whereas local information is more useful for verbs and adjectives. However, the comparison was done using only a single word for each part-of-speech, and it is not clear how representative these are of the general situation for words in that class. Regardless of the part-of-speech, topical information is mostly useful when the difference between senses can be attributed to a specific domain. Senses which are less domain-specific, and more ubiquitous, are not as easily distinguished using topical features.

Availability of Monosemous Words Leacock et al. (1998) state that 64% of the words they examined had monosemous relatives (as provided by WordNet) that were present in the corpus. While this figure is quite high, it is certainly not sufficient for many purposes. Many words and senses may not have any monosemous relatives at all. Also, it is not clear how many of the relatives they reported where closely related words, such as synonyms, and how many were more distant, and therefore presumably less useful. In addition, the corpus frequency of the monosemous relatives was not stated. It is important to note that their method relies on a web-size corpus large enough to contain many examples of the monosemous relatives. While the web certainly fulfils that requirement for English, it is not clear if this is the case for other, rarer, languages. In particular, there is no reason to expect even a large corpus to contain many examples of every monosemous relative, since many may be rare, domain specific, or both. If this is the case, using a smaller corpus than the web may not provide sufficient occurences on which to train a classifier.

Specificity The neighbors provided by a semantic resource are often more accurate, being based on the knowledge of human experts. However, a semantic resource is a stand-alone knowledge base, and is designed to be as general as possible. It is therefore, in many cases, badly suited for use in a specific domain or corpus. It will provide neighbors for all senses, even rare ones, which may appear rarely, or not at all, in our chosen corpus. In addition, it may provide as neighbors words which have a similar sense to the target, but also have more frequent senses, which are more likely to be present in the corpus (see Section 4.3.2). Distributional neighbors, on the other hand, are anchored in the corpus. Although these are often unevenly distributed among the senses of the target, with a strong skew towards the first sense, they are almost always relevant and are guaranteed to be present in the corpus.

Pseudo Labels vs. Predominant Sense Since the data-creation method we present shares some common elements (e.g., the use of distributional and semantic similarity) with the automatic predominant-sense detection algorithm of McCarthy et al. (2004), one might reasonably ask whether there is reason to prefer our method over theirs. Is it not sufficient just to make use of the automatically detected predominant sense for WSD? The answer to this is simple. While McCarthy et al.'s (2004) method focuses on detecting a single predominant sense throughout the corpus, our data-creation method

builds a dataset that allows us to learn about and identify *all* the (prevalent) senses existing in the corpus. Despite the fact that the most-frequent-sense heuristic is a strong baseline, and determining the predominant sense provides a good fallback option in case of limited local information, it does not constitute a true context-specific WSD algorithm. Any approach that ignores local context, and labels all instances with the same sense, providing no information about the secondary senses, has very limited effectiveness when WSD is needed in an application. In addition, such global approaches run the risk of completely mistaking the predominant sense, and thereby mis-labeling most of the instances, whereas approaches that consider local context are less prone to such large-scope errors.

A final issue that needs to be addressed when using any automatic data-creation method is the choice of classifier. Our experiments investigated a selection of standard machine learning classifiers employing different approaches, and can offer some important information as to the choice. We discussed the issue of secondary-sense coverage in Section 4.5.2. The significance of the secondary senses may vary according to the application, and this should be a consideration when choosing which classifier to use. It is also interesting to note that while the Label Propagation algorithm preformed relatively poorly when using the manually labeled data, it ranks very highly when using the automatically labeled data (see Table 4.3). A possible explanation has to do with the nature of the automatically acquired data. The instances in this data are not actual occurrences of the target word, but rather occurrences of similar related words, and therefore have slightly different properties to those in the test set. In addition to learning to distinguish between different senses, it is also important to learn which instances in the training set are closest to a given instance in the test set. The other classifiers we examined deal only with the classes in the training set, whether the focus is on distinguishing between classes, as in the case of SVMs, or on modeling them accurately, as in Maximum Entropy models. The graph based label-propagation method, on the other hand, does not separate the training and test set (it is principally a semi-supervised method). It combines the two datasets, allowing the properties of both to influence the structure of the resulting graph. This suggests that a semi-supervised classifier may be a good choice when using automatically created training data.

4.6.1 Summary

In this chapter, we presented an unsupervised approach to WSD which retains many of the advantages of supervised methods, while being free of the costly requirement for manually-annotated data. It focuses on the data-creation stage, thereby enabling the use of supervised learning techniques for the difficult disambiguation stage. The method makes use of similarity metrics in the distributional and semantic space to provide sense-labeled training data suitable for use with any supervised machine-learning classifier. Our experiments show that the data created by out method produces superior results to that of other data-creation methods in the literature. We also demonstrated that classifiers trained using our method can out-perform state-of-the-art unsupervised methods, and approach the accuracy of fully-supervised methods trained on large amounts of manually-annotated data.

The method we described in this chapter operates under the assumption of the existence of a predefined sense inventory. Under this setting, supervised methods commonly outperform unsupervised ones, and it is therefore desirable to bring supervised methodology into the unsupervised setting. However, the reliance on a fixed list of senses represents a serious obstacle to applied WSD, since the predefined sense distinctions are often unsuitable or irrelevant to the task at hand. In this aspect, unsupervised methods have the advantage. Unlike supervised methods, which are constrained to the set of labels used for training, they can induce the relevant senses directly from the data at hand. This approach has great potential, since it allows unsupervised WSD to be easily integrated into specific applications, and tailored to new tasks and domains, without the need to define a new purpose-built sense inventory and corresponding training dataset. We therefore focus our efforts in this direction in the next chapter.

Chapter 5

Sense Induction with Latent Dirichlet Allocation

5.1 Introduction

In the introduction to this thesis, we noted the importance of unsupervised WSD for new domains and languages. In the previous chapters we examined classic approaches to unsupervised WSD, and presented ways to use lessons learned from supervised methods to relax some of the restrictions of unsupervised approaches and to improve performance. However, one important limitation still remains. From the early days of WSD (e.g., Lesk 1986), unsupervised methods have tended to focus on disambiguation according to, and with the aid of, dictionaries or other lexical resources (see Section 2.1.1). From a pragmatic perspective, there are several strong drawbacks to such an approach. Unsupervised methods are of importance primarily for new domains and languages, where labelled training data is scarce. However, most dictionaries are biased, lacking senses relevant to some domains, while providing definitions for senses that are rare or absent in others. In addition, the granularity of the sense distinctions is fixed, and may not be suitable for the specific task at hand. For new languages, the problem is even more severe, as suitable lexical resources may not exist.

These considerations argue in favor of unsupervised sense induction (or discrimination), where the sense distinctions arise directly from the data, and are therefore more likely to be suitable to the task and domain at hand. There is little risk that an important sense will be left out, or that irrelevant senses will influence the result. Sense induction is applicable to languages which are short on lexical resources, such as comprehensive machine-readable dictionaries. Recent work in machine translation

(Vickrey et al., 2005) and information retrieval (Véronis, 2004) indicates that induced senses can lead to improved performance in areas where methods based on a fixed set of senses have previously failed (Carpuat and Wu, 2005; Voorhees, 1993).

For these reasons, in this chapter we develop a novel approach to sense induction. The task is typically treated as a standard unsupervised clustering problem. However, our approach is different. The sense induction system we present is based on an extension of Latent Dirichlet Allocation (LDA), a probabilistic generative model. The generative approach is well suited for language modelling, and has been successfully employed for many NLP tasks. The model has many of the advantages common in supervised methods, as it has been well studied, and has a variety of available tools and inference techniques. The probabilistic nature of the model allows easy combination with other systems, using mixture or product models. Our extension to the original model provides the means of combining several layers of informative input (e.g., different feature classes), a useful property for many tasks, and a common practice in WSD. Our multi-layer model is general, and can be used for other applications. In the following sections we describe our model in detail, and demonstrate its effectiveness in achieving state-of-the-art performance on the task of sense induction.

5.2 Related Work

5.2.1 Previous Approaches to Sense Induction

Sense Induction is commonly viewed as an unsupervised clustering problem, where instances of a target word are partitioned into classes by considering their contexts. There are many approaches regarding how to address the clustering problem. In this section we present an overview of a selection of methods representing different clustering approaches for sense induction. They vary as to representation, problem formulation, choice of feature space, and how they address the issue of model order (optimal number of clusters).

Clustering by Committee Pantel and Lin (2002) present the *Clustering by Committee* (CBC) algorithm which employs a distributional-similarity metric over a vector representation. The algorithm clusters together words sharing an induced sense. Each sense cluster is represented by a small sub-group of words which are very similar to one another (the committee). Each word is represented by a (sparse) vector of de-

pendency co-occurrence counts. For instance, the word *ball* might have the feature *object-of-throw* with a value related to the number of times it occurred as the object of the verb *throw* in a given corpus. The actual value the authors use is the mutual information between the target and dependent words (*ball* and *throw*, in our example). Similarity is measured with the help of a *Distributional Similarity* metric defined over these dependency vectors (for more information, see Section 4.3.1). The dependency features provide a strong characterization of the word, and similarity in this feature distribution space has been shown to correlate strongly with semantic similarity between words (Lin, 1998b).

Clustering is performed in a three-phase procedure. First, for each word, the top-ten most similar words are calculated. These are clustered, and the "tightest" cluster (highest average pairwise similarity between members) is added to a list of candidate committees. In the second phase, proceeding iteratively from the highest ranking group in the candidate list, the group centroid is calculated and compared to centroid of each of the set of existing clusters. If it is different enough (similarity is below a certain threshold), the candidate group becomes the committee of a new cluster, and is added to the existing ones. The committees are now fixed, and their centroid represents the cluster as a whole. Any elements added to the cluster in the next phase do not effect the cluster centroid. In the third phase, any unassociated words are added to the most similar cluster. Once a word has been added to a cluster, the feature vector representing that word is stripped of features overlapping with those of the cluster centroid. The stripped vector (representing a possible less-frequent sense of the word) is again evaluated against the existing clusters. This is repeated until the stripped vector is not similar to any cluster (below a specific threshold). The authors performed both an automatic comparison of their clusters to WordNet and a manual evaluation, reporting 63% and 72% accuracy, respectively.

Hyperlex The Hyperlex algorithm (Véronis, 2004) addresses the clustering problem from a graph-based perspective. It makes use of the "small-world" properties of co-occurrence graphs and detects "hubs" in the graph which represent induced senses. For each target word, a set of context paragraphs containing the word are retrieved from the web using the plural and singular form of the word as a query. These form the dataset for each word. A graph is constructed with a node for every noun and adjective occurring in the dataset more that five times. Edges link any two words which co-occur in the same paragraph, and the weight of the edges, representing semantic distance, is

set to $w = 1 - \max(p(A|B), p(B|A))$, where the probabilities are estimated from the word frequencies. A simple algorithm detects and separates the hubs in the graph by iteratively finding the most frequent word, converting it to a hub, and removing it and its neighbors from the graph. The algorithms stops when the new hubs no longer meet certain specifications (related to frequency and number of neighbors). Once the hubs (set of senses) are determined, each word in the graph is assigned a score vector which is zero for all indexes except that of the closest hub, where the value is the distance between the word and the hub. In order to disambiguate the target word in a given context, the score vectors of all the words appearing in the context are summed, and the sense (hub) with the highest score is chosen. Manual evaluation was performed, with the hub-senses being represented by the ten words closest to each. The algorithm achieved very high (95.5%) accuracy in that setting.

I2R One of the best performing systems on the Semeval sense induction task, I2R (Niu et al., 2007), takes an information-theoretic perspective. It makes use of the Sequential Information Bottleneck (sIB) algorithm (Slonim et al., 2002), to cluster the instances into a predetermined number of clusters. The sIB algorithm is based on an information-theoretic formulation, and views the clustering task as an optimization problem. It attempts to group together values of one variable while retaining as much information as possible regarding the values of another (target) variable. There is a trade-off between the compactness of the clustering and the amount of retained information, known as the *Information Bottleneck*. In the sense induction setting, the algorithm is used to cluster together instances which have similar feature distributions (the target variable), thus reducing the amount of information lost by merging several instances into a single cluster. The sequential IB algorithm works iteratively. Starting with a random partition of the instances into the designated number of clusters, it sequentially draws each of the elements from its current cluster, and places it in the cluster to which it is most similar, i.e., for which the merging cost (in terms of information loss) is lowest. The algorithm proceeds until it reaches a local maximum where each instance is most similar to its current cluster. The feature space used by the authors contained parts of speech of neighboring words with position information, unordered single words in the context, and local collocations. No syntactic relations were used. Rather than determining the required number of clusters via heuristic manually-specified thresholds, as in the previous methods, the authors address the issue of model-order through a cluster-validation procedure. For every value of K between 2 and 5,

clustering is performed on the original dataset, and on twenty random subsets containing 90% of the instances. A scoring function is defined, based on how many of the instance pairs in the random subsets maintain the same/different-cluster association as in the full dataset. The value of K which maximizes this function is chosen.

Though all the methods we mention view the problem from a clustering perspective, they differ in several respects. Two of the methods address the clustering task using different forms of vector similarity (information-theoretic, as in I2R, or semantic, as in CBC), while the third sees it as a matter of detecting high density areas in graphs. The first method we mentioned, CBC, uses an all-word setting, while the other two handle each ambiguous word individually. The feature representation used can be highly specific syntactic information, requiring an accurate dependency parser (CBC), more general collocation information (Hyperlex), or a tailored feature set, somewhere in between (I2R). Finally, the important issue of model order is addressed differently in each case (through heuristic thresholds in CBC and Hyperlex, and with a cluster-validation procedure in I2R).

5.2.2 LDA Topic Models

Our treatment of the sense induction problem differs from the standard clustering approach. We base our system on a generative probabilistic graphical model, Latent Dirichlet Allocation (LDA), first proposed by Blei et al. (2003) for modelling text generation. The model posits that each document is generated by selecting a distribution of topics from a family of parametrized Dirichlet distributions. The words in the document are then generated by repeatedly sampling a topic according to the topic distribution, and selecting a word given the chosen topic. For the purpose of inference, the model is reversed, and the most likely topic distribution and word assignments are calculated from the observed data. The generative nature of the model allows it to handle newly observed documents which do not conform precisely to a previously seen distribution.

Figure 5.1 presents an example of the output of LDA on a single document (taken from Blei et al. 2003). Each word is assigned a single topic, and the overall topic distribution is inferred from the distribution of these assignments. The topics which pertain to the example document are shown in Figure 5.2. For each topic, a list of the most probable words for that topic is shown. The topic headings are not provided by the LDA system, but were manually added by the authors for clarification.

The$_4$ William Randolph$_3$ Hearst$_3$ Foundation$_2$ will give \$1.25$_2$ million$_2$ to Lincoln$_1$ Center$_3$, Metropolitan Opera$_1$ Co.$_3$, New$_1$ York$_1$ Philharmonic$_1$ and Juilliard School$_4$. "Our board$_2$ felt that we had a real opportunity$_3$ to make$_3$ a mark$_1$ on the future$_3$ of the performing$_1$ arts with these grants$_2$ an act$_1$ every bit$_1$ as important$_3$ as our traditional$_3$ areas of support$_2$ in health, medical research$_2$, education$_4$ and the social services$_2$," Hearst$_3$ Foundation$_2$ President$_2$ Randolph$_3$ A. Hearst$_3$ said Monday$_4$ in announcing$_2$ the grants$_2$. Lincoln$_1$ Centers share$_3$ will be \$200,000$_2$ for its new$_1$ building$_2$, which will house$_2$ young$_3$ artists and provide$_2$ new$_1$ public$_2$ facilities$_2$. The Metropolitan Opera$_1$ Co. and New$_1$ York$_1$ Philharmonic$_1$ will receive$_2$ \$400,000$_2$ each. The Juilliard School$_4$, where music$_1$ and the performing$_1$ arts are taught$_4$, will get \$250,000$_2$. The Hearst$_3$ Foundation$_2$, a leading$_1$ supporter$_1$ of the Lincoln$_1$ Center$_3$ Consolidated Corporate Fund$_2$, will make$_3$ its usual annual$_2$ \$100,000$_2$ donation, too.

Figure 5.1: Example output of the LDA model on a single document. Each word is assigned a single topic, indicated by its subscript, and referring to the topics listed in Figure 5.2.

1."Arts"	2."Budgets"	3."Children"	4."Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Figure 5.2: Topics pertaining to the example document, along with a list of most probable words for each topic. Topic headings were added manually by Blei et al. (2003).

The authors present efficient approximate inference techniques over the model (see Section 5.3.2). They also compare to other models proposed in the literature, and report improved results on document modelling and text classification tasks, where their model overfits considerably less than the others. The authors also emphasize the applicability of the model to other areas where the data can be modelled in a similar fashion, using repeated-structures and underlying distributions, and demonstrate by applying it to a collaborative filtering task (user/movie preference).

Recently, there have been several attempts to make use of LDA topics for WSD. Boyd-Graber and Blei (2007) propose a model that takes an all-words approach and integrates the original LDA process for document modelling with the idea of using distributional neighbors, as proposed in McCarthy et al. (2004) (see Section 3.2.3.2). As in the original LDA model, documents are generated word by word. For each word, a topic is sampled from the document's topic distribution, and a word is generated from that topic. In addition, a distributional neighbor is selected based on the topic and distributional similarity to the generated word. The authors specifically design the model so that the addition of neighbors does not effect the topic assignment procedure. This enables them to use the variational inference method described in Blei et al. (2003) to acquire topic assignment probabilities for each word. These assignments are used to calculate the most-probable sense given the neighbor, using a topic-specific version of the Jiang-Conrath (Jiang and Conrath, 1997) semantic similarity measure. The authors evaluate their method on several datasets and tasks, but the results do not achieve significant improvement over the respective baselines, and do no better than the original method of McCarthy et al. (2004).

In a supervised setting, Cai et al. (2007) replace the common bag-of-words document representation by a bag-of-topics one, using topics derived from an LDA model. This helps simple algorithms, such as Naive Bayes, by reducing the sparsity of the vector space. When using more powerful algorithms, such as SVM, which can handle high-dimensional sparse data, the benefit is smaller.

A key element in these previous attempts at using LDA for WSD is the tendency to remain at a topic-based, document-like setting. When modelling text, LDA posits that each word is generated from a specific topic. When dealing with documents, these LDA topics often resemble high-level categories used to describe the subject matter, such as 'arts' and 'education' (Blei et al., 2003). While such categories can be useful for document classification, and have been shown to be a useful source of additional information for WSD methods (see Leacock et al. 1998; Bordag 2006), they are in-

sufficient on their own. It is unreasonable to expect a single set of high-level topics to differentiate between every two senses of every ambiguous word. Furthermore, LDA need not be restricted to the topic-modeling setting. The model is general, and can be used to model other types of grouping inherent in the data, such as genre, sentiment or word-senses. In this respect, LDA can be viewed as a general-purpose form of clustering algorithm, based on a generative, probabilistic, model. LDA has several advantages over standard unsupervised clustering techniques. It is a probabilistic model, and thus inherently modular. As we have shown in Chapter 3, modularity is an important prerequisite for WSD. Since such models specify probability distributions over possible values, they are easy to integrate and combine with each other as mixture or product models. The LDA generative approach is appropriate for modeling language, where a latent structure in the speaker's mind is responsible for generating words[1]. In addition, LDA has many of the advantages of supervised techniques, as it has been widely studied, and comes with a variety of standard tools and inference techniques. It has been used in many natural language processing tasks besides WSD. Examples include entity correference resolution (Bhattacharya and Getoor, 2006) and part-of-speech tagging (Toutanova and Johnson, 2008).

In this chapter, we present an LDA-inspired model specifically designed to handle the sense induction problem directly. As discussed in Section 3.5, our approach in this thesis is to handle each ambiguous word individually. We therefore create a separate disambiguation model for every target word, while employing a small number of sense-clusters meant to capture the possible senses of that word. This is in marked contrast to the tens, and sometimes hundreds, of topics commonly used in document-modeling tasks. We also make use of much smaller units of text (a few sentences, rather than a full document), in order to focus on local sense-clusters, rather than high-level topical information. As we are dealing with sense *induction*, we do not rely on a pre-existing list of senses, and do not assume a correspondence between our automatically derived sense-clusters and those of any given sense inventory. Such a mapping is only performed when necessary to enable evaluation and comparison with

[1] Although several clustering models, such as Gaussian Mixture Models (GMM), can be considered probabilistic and generative, they differ significantly from the LDA approach. Clustering models are generative, since they attempt to provide a representation of the underlying classes (clusters) in the data, as apposed to discriminative methods, which seek only to distinguish between them. However, they do not take into account the linguistic process, or attempt to model specifically the generation of text. LDA considers these aspects, and presents a formulation which is well suited to the natural properties of language, such as the Zipfian distribution of words. While these factors can be introduced into other clustering models in the form of priors and special features, this requires careful engineering, while in the LDA framework they arise naturally as part of the model.

other disambiguation methods.

The model we develop is enhanced to allow the integration of several information layers. In many tasks, we have several useful sources of information about the object of interest. For instance, for the purpose of document classification, both text and images in the document are of interest, since each of these can provide information about the document. In speech summarization, layers of speech-based features can provide a helpful addition to the transcribed text (see Murray et al. 2006). In WSD, researchers have long been combining information from different sources, such as lexical resources, grammatical information and context information (see, for example, Preiss 2004, and Florian et al. 2002). Our own work (see Chapter 3) also shows the benefit of integrating multiple sources of information in the task of unsupervised WSD. We would, therefore, like to adapt the generative LDA model to allow such integration. The modified version of LDA which we present (Layered-LDA) provides this utility.

Few works have addressed the issue of integrating multiple information sources in the LDA framework. Griffiths et al. (2005) present a composite model that integrates document-level contextual information with short-range syntactic dependencies for the purpose of document modelling. Their model combines the classic LDA model, which addresses high level topics, with an HMM for modelling local dependencies. The resulting model is competitive on the tasks of part-of-speech tagging and document classification. Barnard et al. (2003) present several models for automatically annotating images with description keywords. Their basic LDA approach (which they call multi-modal LDA or MoM-LDA) is similar to a two-layer (text and images) version of our system. It assumes independence among the image and text layers, but uses information from both. The more sophisticated models presented in that paper attempt to do away with the independence assumption and focus on the main task they address - jointly modeling image fragments and descriptive keywords. Both these approaches focus on a specific setup and task (long- and short-range contextual information in the first instance, text and images in the second), and tailor their models appropriately. They therefore do not represent a general solution to the problem of combining multiple information sources. Our approach, on the other hand, is designed as a general extension of the LDA model. It is not application-specific, and can be used for any task where multiple layers of information exist.

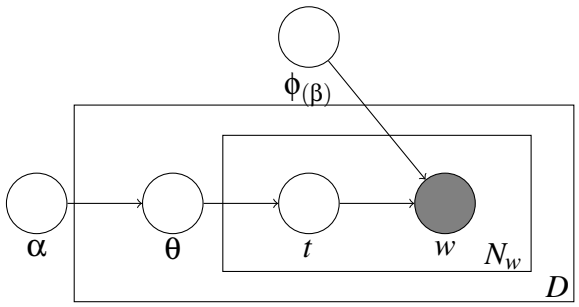

Figure 5.3: LDA model for topic-based document modelling. Shaded nodes represent observed variables. Rectangles represent repeated structures. The outer rectangle represents a document (D), and the inner one represents the choice of a topic (t) and word (w), repeated for each of the N_w words in the document.

5.3 The Model

5.3.1 Sense Induction

As mentioned in Section 5.2.2, the original LDA model (represented in Figure 5.3) posits that each document is generated by selecting a distribution (θ in the figure) of topics from a Dirichlet distribution parametrized by α. The words in the document are generated by repeatedly sampling a topic according to the topic distribution, and selecting a word given the chosen topic according to the word-topic distribution ϕ (parametrized by β).

We propose to adapt the classic LDA model to our WSD task by making several changes in the original document-generation model[2]. We are not interested in modelling a whole document as a collection of words produced by a distribution of high-level topics. Instead, we wish to present the local context surrounding a single instance of an ambiguous target word as a collection of context elements produced by a distribution of senses of the word (see Figure 5.4). Context elements may be any sort of relevant information, such as nearby words, part-of-speech information and so on. We describe the full set of elements with which we experimented in Section 5.4.2.

In a simple example case, where context elements are words, and each context is a 20-word window centered around the ambiguous target word, the generative process is as follows. A distribution θ over the possible senses of the target is sampled from

[2]The original LDA software on which we based our model is GibbsLDA++, a C/C++ Implementation of Latent Dirichlet Allocation (LDA) using Gibbs Sampling for Parameter Estimation and Inference, by Xuan-Hieu Phan. Available at `http://gibbslda.sourceforge.net/`.

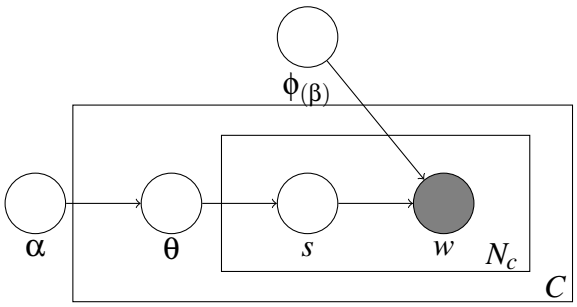

Figure 5.4: Modified LDA model for word-sense modelling. Shaded nodes represent observed variables. Rectangles represent repeated structures. The outer rectangle represents the context C of an instance, and the inner ones represent repeated choice of a sense (s) and a context word (w), repeated for each of the N_c words in the context.

a family of Dirichlet distributions parametrized by α. Then, for each of the twenty words in the window, a sense is sampled from the sense-distribution, and the word is generated given that sense assignment, according to the multinomial word-sense distribution ϕ (parametrized by β).

As mentioned above, we wish to make use of a richer context representation, containing several categories of features (not only a single 20-word window). We therefore enhanced our version of the LDA model with the ability to deal with several feature layers. Figure 5.5 shows a symbolic representation of our layered model. We have multiple layers of information, each composed of a different class of features. For instance, one layer could contain the words observed in a 20-word window, representing high-level topical information. Another layer could contain part-of-speech bigrams adjacent to the target, and represent syntactic information. The full list of feature classes used in our experiments is detailed in Section 5.4.2. Information from all of the layers is combined when estimating the sense distribution of each instance.

Under the layered model, for each instance, each layer is generated in a similar fashion to the single word-window layer described above. For each element in the layer, in turn, a sense assignment is sampled from the sense distribution θ, which is shared by the whole instance. Then, a value is sampled for the element (a word in the word-window layer, a part-of-speech bigram in the PoS-bigram layer, etc.), given the sense-assignment, from the appropriate multinomial distribution for that layer (ϕ_j).

The model operates under the simplifying assumption of independence between the layers. Many probabilistic models assume independence between multiple sources of information, to reduce computational complexity, despite the fact that such inde-

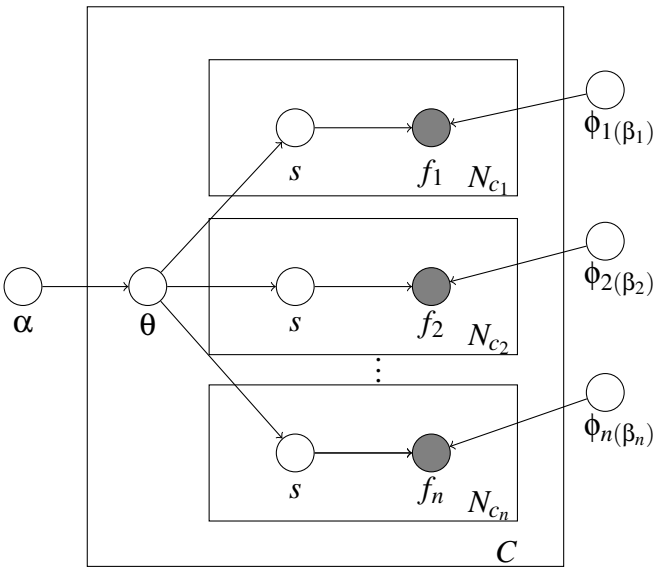

Figure 5.5: Extended sense induction model; inner rectangles represent different sources (layers) of information. All layers share the same, instance-specific, sense distribution (θ), but each have their own (multinomial) sense-feature distribution (ϕ). Shaded nodes represent observed features f_i; these can be words, parts of speech, collocations or dependencies.

pendence is rarely present in reality. The classic model for probabilistic classification is the Naive Bayes method, so named for exactly this reason. Despite their simplicity, such models achieve high performance in many tasks (see Langley et al. 1992).

5.3.2 Inference

Several inference approaches have been proposed for LDA models. Exact inference is intractable, given the complexity of the model. Instead, various approximate inference techniques have been suggested. Blei et al. (2003) propose an EM-based maximum likelihood inference technique, using a variational E-step. Goldwater (2007) reviews several common inference techniques and their limitations. She then describes Markov Chain Monte-Carlo (MCMC) sampling algorithms and in particular the Gibbs sampling algorithm (Geman and Geman, 1984) which can be used to estimate the posterior distribution of the parameters, as well as the *maximum a posteriori* (MAP) or expected value thereof. The Gibbs sampling procedure is iterative. In each iteration, a value is sampled for each variable in the model in turn, according to the conditional probability given the current values of all the other variables.

5.3.3 Notation

Before formally describing our model, it is neccessary to present and explain the notation we use. This are summarized in Table 5.1. To begin with, our model contains a number of constants. K indicates the number of sense-clusters used in the model. This is essentially another hyperparameter, which is provided in advance (see discussion in Section 5.5.2). We use L to represent the number of different input layers in the data and V_l to indicate the size of the vocabulary in each layer, i.e., the number of different values which can occur in the layer (for example, in the word window layers, V_l is equal to the number of different word types in the data).

We also make use of several levels of variables and parameters. First, there are the hyperparameters and top level variables, which are the most global. The variable α parametrizes the Dirichlet distribution of senses in each instance. Specifically, it controls the probabilities of the family of possible sense-distributions from which is sampled the individual sense distribution of each instance. In our mathematical analysis, we decompose α into portions α_l, one for each layer. One interpretation of α is as a pseudo-count prior, or smoothing factor, which provides weight in addition to that which was actually observed in the data. Under this interpretation, we can speak of α_l as being the portion of the pseudo-count that is given to each input layer l.

For indexes, we use m to index the documents, and l to index the layers. The index i is used to distinguish features and their associated senses. This index is global and ignores document and layer boundries[3].

The second category of variables relates to individual instances. Each instance has a sense distribution θ, representing the relative portion of each sense in that instance (in terms of the sense assighnments of the individual features comprising that instance).

Each instance is made up of observed features $\{f_i, f_{i+1}, ...\}$ and their corresponding latent sense assignments $\{s_i, s_{i+1}, ...\}$. The collection of all the observed features in the data (all instances) is indicated by $\overline{f}$, and similarly, $\overline{s}$ represents the collection of all the sense assignments. We use $\overline{f}_{-i}$ to indicate all features except the current one (f_i), and $\overline{s}_{-i}$ for all sense assignments except s_i.

In our notation $\#(x)$ indicates the number of times the event x was observed in the data (all instances). Similarly, for a specific instance, $\#m(x)$ represents the number of times event x was observed in document m, and $\#m$ is the total number of events in document m, i.e., the size of the document. On the layer level, $\#l(x)$ represents

[3] An alternative would be to provide a triple index for each feature and its sense assignment, indicating the document, layer and location in the layer. We chose a single index for simplicity.

the number of times event x was observed in layer l (of a certain document), and $\#l$ indicates the size of the layer.

5.3.4 Model Formulation

In order to formally describe our model, we must outline the underlying probabilistic assumptions, and derive the update function used in the Gibbs sampling procedure, i.e., the conditional distribution of a single assignment given the current assignments of all the other variables. In our model, each element in each of the layers (e.g., each word in the ± 10-word window, or each part-of-speech bigram in their layer) is a variable, and is assigned a sense label. From these assignments, the sense distribution of the instance as a whole can be determined. We need to provide the conditional probability of the i-th variable (for example, the part-of-speech bigram preceding the target) being assigned sense label s_i, given the feature-value f_i of the variable (e.g., the $\langle verb, determiner \rangle$ bigram value), and the current sense assignments of all the other variables in the data ($\overline{s}$).

We begin with the basic Bayesian formulation. The probability of a single sense assignment, s_i, is proportional to the product of the likelihood (of the feature-value f_i of the i-th variable, given the rest of the data) and the prior probability of the assignment.

$$p(s_i|\overline{s}_{-i}, \overline{f}) \propto p(f_i|\overline{s}, \overline{f}_{-i}, \beta) \cdot p(s_i|\overline{s}_{-i}, \alpha) \tag{5.1}$$

For the likelihood term, integrating over all possible values of the multinomial feature-sense distribution ϕ gives us the rightmost term in Equation 5.2.

$$p(f_i|\overline{s}, \overline{f}_{-i}, \beta) = \int p(f_i|l, \overline{s}, \phi) \cdot p(\phi|\overline{f}_{-i}, \beta_l) d\phi = \frac{\#(f_i, s_i) + \beta_l}{\#(s_i) + V_l \cdot \beta_l} \tag{5.2}$$

This term has an intuitive interpretation. The notation $\#(f_i, s_i)$ indicates the number of times the feature-value s_i was assigned sense s_i in the rest of the data. Similarly, $\#(s_i)$ indicates the number of times the sense assignment s_i was observed in the data. β_l is the Dirichlet prior for the feature-sense distribution ϕ in the current layer, and V_l is the size of the vocabulary of that layer, i.e., the number of possible feature values in the layer. Intuitively, the probability of a feature-value given a sense is directly proportional to the number of times we've seen that value and that sense-assignment together in the data, taking into account a pseudo-count prior, expressed through β.

A similar approach is taken with regards to the prior probability. In this case,

Constants	
K	number of senses
L	number of layers
V_l	size of the vocabulary (number of types) in layer l
Global	
α	hyperparameter of the Dirichlet sense distribution family
ϕ	joint word-feature distribution (multinomial)
β	hyperparameter of the word-sense distribution
Indexes	
m	document index
l	layer index
i	feature and sense assignment index (global)
Instance	
θ	sense distribution for a specific instance
f_i	the i-th feature
s_i	sense assignment for the i-th feature
$\overline{f}$	the collection of all features in the data
$\overline{f}_{-i}$	the collection of all features in the data except f_i
$\overline{s}$	the collection of sense assignments of all features in the data
$\overline{s}_{-i}$	the collection of sense assignments of all features in the data except s_i
Count Notation	
$\#(x)$	number of times event x was observed in the data (all instances)
$\#m(x)$	number of times event x was observed in instance m
$\#m$	total number of events in instance m (size of the instance)
$\#l(x)$	number of times event x was observed in layer l (in a specific instance)
$\#l$	size of layer l (in a specific instance)

Table 5.1: Notation used in description of the layered LDA model

however, all layers of information in the instance must be considered.

$$p(s_i|\overline{s}_{-i},\alpha) = \sum_l \lambda_l \cdot p(s_i|l,\overline{s}_{-i},\alpha_l) \tag{5.3}$$

Here λ_l is the weight for the contribution of layer l, and α_l is the Dirichlet prior for the sense distribution θ in the current layer. Treating each layer individually, we integrate over the possible values of θ, obtaining a similar count-based term.

$$p(s_i|l,\overline{s}_{-i},\alpha_l) = \int p(s_i|l,\overline{s}_{-i},\theta) \cdot p(\theta|\overline{f}_{-i},\alpha_l) d\theta = \frac{\#l(s_i)+\alpha_l}{\#l+S\cdot\alpha_l} \tag{5.4}$$

$\#l(s_i)$ indicates the number of elements in layer l assigned the sense s_i, $\#l$ indicates the number of elements in layer l, i.e., the size of the layer (in the current instance). S is the number of senses. Here, too, the intuitive interpretation is that the prior for sense s_i in a specific layer is its observed proportion in that layer, taking into account the pseudo-count α_l.

To distribute the pseudo counts represented by α in a reasonable fashion among the layers, we define $\alpha_l = \frac{\#l}{\#m}\cdot\alpha$ where $\#m = \sum_l \#l$, i.e., the total size of the instance. This distributes α according to the relative size of each layer in the instance.

$$p(s_i|l,\overline{s}_{-i},\alpha_l) = \frac{\#l(s_i)+\frac{\#l}{\#m}\cdot\alpha}{\#l+S\cdot\frac{\#l}{\#m}\cdot\alpha} = \frac{\#m\cdot\frac{\#l(s_i)}{\#l}+\alpha}{\#m+S\cdot\alpha} \tag{5.5}$$

Placing these values in Equation 5.3 we obtain the equation specifying the overall prior probability, which is a simple weighted average of the priors from the individual layers.

$$p(s_i|\overline{s}_{-i},\alpha) = \frac{\#m\cdot\sum_l \lambda_l\cdot\frac{\#l(s_i)}{\#l}+\alpha}{\#m+S\cdot\alpha} \tag{5.6}$$

Putting it all together, we arrive at the final update equation for the Gibbs sampling:

$$p(s_i|\overline{s}_{-i},\overline{f}) \propto \frac{\#(f_i,s_i)+\beta_l}{\#(s_i)+V_l\cdot\beta_l}\cdot\frac{\#m\cdot\sum_l\lambda_l\cdot\frac{\#l(s_i)}{\#l}+\alpha}{\#m+S\cdot\alpha} \tag{5.7}$$

Note that when dealing with a single layer, this equation collapses to Equation 5.8, which is identical to the Gibbs update equation for the original LDA algorithm (with $\#m(s_i)$ indicating the number of words in the document assigned to sense-cluster s_i).

$$p(s_i|\overline{s}_{-i},\overline{f}) \propto \frac{\#(f_i,s_i)+\beta}{\#(s_i)+V\cdot\beta}\cdot\frac{\#m(s_i)+\alpha}{\#m+S\cdot\alpha} \tag{5.8}$$

The sampling algorithm gives direct estimates of s for every context element. However, in the context of our task, we are more interested in estimating θ, the sense-context distribution. This can be obtained as in Equation 5.6, but taking into account all sense assignments, without removing assignment i.

5.4 Experimental Setup

5.4.1 Data

For evaluation, we used the dataset provided in the sense induction and discrimination task in Semeval-2007 (Agirre and Soroa, 2007). This is comprised of text from the Penn Treebank II (sections 1 and 22-24 were used for test data, and the rest for mapping, see Section 5.4.3). The Treebank data is a collection of articles from first half of the 1989 Wall Street Journal. Table 5.2 shows some of the properties of the Semeval dataset (both portions combined). The average ambiguity is approximately four senses, with a high (almost 80%) skew towards the predominant sense. This means that an algorithm which simply chooses the most frequent sense of each word to label all the instances achieves almost 80% accuracy. This skew is partly the result of the fact that OntoNotes (Hovy et al., 2006) senses were used in Semeval, instead of the finer-grained WordNet ones. Coarser senses make the inference task easier, but also make it very difficult to beat the first-sense baseline.

For our experiments, we used two learning corpora. The British National Corpus (BNC) served as our out-of-domain corpus, and contained approximately 730 thousand instances of the 35 target nouns in the Semeval lexical sample. The second, in-domain, corpus was built from selected portions of the Wall Street Journal (WSJ) corpus. We used all articles[4] from the years 1987-89 and 1994 to create a corpus of similar size to the BNC, containing approximately 740 thousand instances of the target words.

A simple way to judge whether two pieces of text share a similar domain is to examine the frequency of occurrence of different words in the data. We can measure divergence between these distributions to determine how (dis-)similar they are. Figure 5.6 shows the distribution of the Semeval target words in the BNC, the WSJ, and the provided test data. The Jensen-Shannon divergence between the instance distribution in the WSJ and Semeval is 0.0166 bits, whereas the divergence between the BNC distribution and that of Semeval is 0.15 – almost ten times as large. This indicates that the WSJ text is much more similar to the test data than the BNC. This is to be expected, as the Semeval data is itself a portion of the WSJ.

The LDA framework contains several parameters whose values must be specified (see Section 5.5.2). We used the Senseval 2 lexical sample (Preiss and Yarowsky, 2001) data as a tuning set, to get an estimate of the desired value for the α parameter.

[4]Excluding the portion used for the Penn. Treebank II, i.e., the Semeval dataset

Word	# Instances	Ambiguity	1st Sense
area	363	3	266 (73.2%)
authority	111	4	53 (47.7%)
base	112	5	40 (35.7%)
bill	506	3	340 (67.1%)
capital	335	4	313 (93.4%)
carrier	132	8	101 (76.5%)
chance	106	4	52 (49.0%)
condition	166	2	135 (81.3%)
defense	141	7	41 (29.0%)
development	209	3	159 (76.0%)
drug	251	2	163 (64.9%)
effect	208	3	169 (81.2%)
exchange	424	5	306 (72.1%)
future	496	3	395 (79.6%)
hour	235	4	201 (85.5%)
job	227	3	172 (75.7%)
management	329	2	205 (62.3%)
move	317	4	295 (93.0%)
network	207	3	123 (59.4%)
order	403	7	336 (83.3%)
part	552	4	441 (79.8%)
people	869	4	791 (91.0%)
plant	411	2	360 (87.5%)
point	619	9	458 (73.9%)
policy	370	2	296 (80.0%)
position	313	7	95 (30.3%)
power	298	3	150 (50.3%)
president	1056	3	887 (83.9%)
rate	1154	2	979 (84.8%)
share	3061	2	2989 (97.6%)
source	187	5	69 (36.8%)
space	81	5	43 (53.0%)
state	689	3	570 (82.7%)
system	520	5	284 (54.6%)
value	394	3	357 (90.6%)
total/avg	15852	3.94	12634 (79.7%)

Table 5.2: Number of instances and ambiguity of each noun in the Semeval lexical sample. The rightmost column presents the number (and percentage) of instances of the word labeled with the most frequent sense.

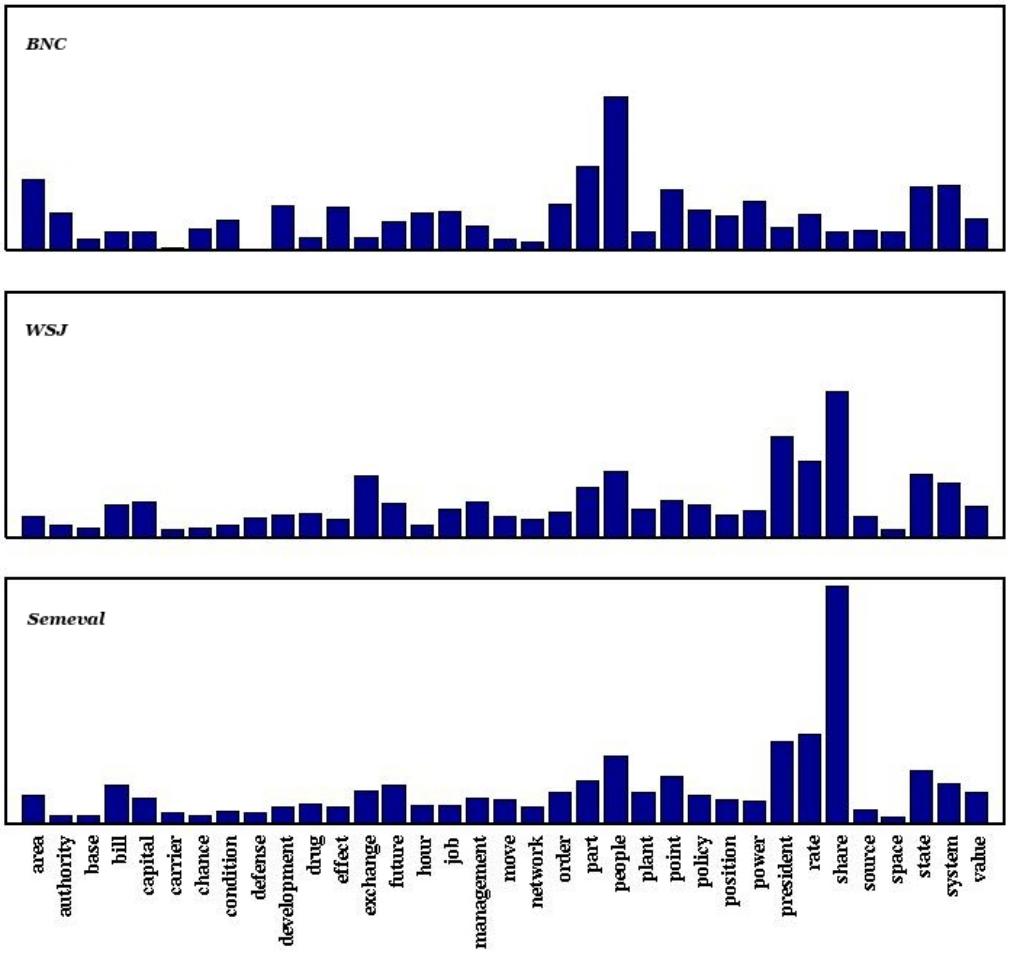

Figure 5.6: Distribution of word occurrences in the two corpora and the evaluation dataset. Each bar represents one of the 35 words in the lexical sample, and the height of the bar indicates the relative frequency (% of instances) in the dataset.

5.4.2 Context Features

For the purpose of our experiments, we used the feature set described in detail in Section 4.4.3. This feature set is designed to capture both immediate local context, wider context and syntactic context. It contains six feature categories: $\pm$10-word window (10w), $\pm$5-word window (5w), collocations (1w), word n-grams (ng), part-of-speech n-grams (pg) and dependency relations (dep). Each feature category is treated as a separate information layer in our model.

The Semeval workshop organizers provided a small amount of context for each instance (usually a sentence or two surrounding the sentence containing the target word). This context, as well as the text in the training corpora, was parsed using RASP (Briscoe and Carroll, 2002), to extract part-of-speech tags, lemmatized forms of the words, and dependency information. For instances containing more than one occurrence of the target word, we chose the first occurrence as the target. Instances which were not correctly recognized by the parser (e.g., a target word labeled with the wrong lemma or part-of-speech), were automatically assigned to the largest sense-cluster.[5]

5.4.3 Evaluation

Because every sense induction system uses its own set of arbitrary labels, evaluation and comparison between different systems is a difficult problem. Several possible solutions have been suggested. One alternative is to manually evaluate and decide on the correctness of each system's clustering solution. This presents several problems. First and foremost, such evaluation is extremely expensive in terms of manual labor, and requires individual evaluation of each system or system variation. Also, in order to make the evaluation fair, the same evaluator should judge all systems. This makes it difficult to introduce and evaluate a new system without re-evaluating previous ones. In addition, it is impossible to make the evaluation standardized, and there is no guarantee that two judges (or the same judge, at different times) will make identical, or even similar, decisions.

Another option is via integration with a particular application (e.g., information retrieval, Schütze 1998), and comparison of the effects of the system on the results. While this approach is pragmatic, it presents many problems of its own. For instance, it is necessary to decide upon a task and application which will allow such integration, and not be biased towards a particular approach or system. In addition, the effects

[5] Less than 1% of the instances

of the integration process and compatibility with the application must be taken into consideration when evaluating system performance.

Another approach attempts to perform the evaluation independently of a specific task, as is commonly done for supervised systems. The proponents of this approach attempt to devise a reasonable way of mapping the results of the unsupervised system to the gold-standard annotation. Pantel and Lin (2002) automatically map induced senses to WordNet, and then manually evaluate the mapping quality. More recently, tagged corpora have been used to map the induced senses to gold standard ones (Purandare and Pedersen, 2004; Niu et al., 2005a; Agirre et al., 2006). This approach allows for standardized evaluation and comparison, but introduces noise arising from the mapping process, and may be biased towards certain types of systems (Agirre and Soroa, 2007).

A final option is to treat the problem as a standard unsupervised clustering task. In this approach, the gold-standard senses represent the true classes (but the labels are ignored), against which the system's clustering solution is evaluated using standard measures from the clustering literature, such as *purity*, *entropy* and *F-score*.

Given a set of gold standard classes (senses) $\{S_i\}_1^n$ and a clustering solution $\{C_j\}_1^m$, and assuming there are $|D|$ instances in the dataset, the *purity* of a cluster C_j is defined by:

$$Purity(C_j) = \frac{1}{|C_j|} \cdot \max_i |C_j \cap S_i|$$

In words, the purity of a cluster is the proportion of the cluster shared with the most similar gold-standard class. The *purity* of the entire clustering solution is:

$$Purity(\{C_j\}_1^m) = \sum_{j=1}^{m} \frac{|C_j|}{|D|} \cdot Purity(C_j)$$

The *Entropy* measure employs a similar approach, but uses an information-theoretic weighting. The *Entropy* of a cluster is defined as:

$$Entropy(C_j) = -\frac{1}{\log m} \cdot \sum_{i}^{n} \frac{|C_j \cap S_i|}{|C_j|} \log \frac{|C_j \cap S_i|}{|C_j|}$$

The *Entropy* of the entire clustering solution is:

$$Entropy(\{C_j\}_1^m) = \sum_{j=1}^{m} \frac{1}{m} \cdot Entropy(C_j)$$

F-Score is similar to that used in information retrieval, assuming the S_i are the correct documents given the query, and C_j are those retrieved by the system. Therefore,

$$Precision(S_i, C_j) = \frac{|C_j \cap S_i|}{|C_j|} \qquad Recall(S_i, C_j) = \frac{|C_j \cap S_i|}{|S_i|}$$

$$F\text{-}Score(S_i, C_j) = \frac{2 \cdot Precision(S_i, C_j) \cdot Recall(S_i, C_j)}{Precision(S_i, C_j) + Recall(S_i, C_j)}$$

$$F\text{-}Score(S_i) = \max_{C_j} F\text{-}Score(S_i, C_j)$$

The *F-Score* of the entire clustering solution is given as:

$$F\text{-}Score(\{C_j\}_1^m) = \sum_{i=1}^{n} \frac{|S_i|}{|D|} \cdot F\text{-}Score(S_i)$$

In the sense induction and discrimination task in Semeval-2007 (Agirre and Soroa, 2007), the task organizers presented a standardized framework for evaluation of unsupervised systems under the latter two approaches described above[6]. They provided a cluster-based evaluation system, which did not attempt to match the induced sense categories with the labels of the gold standard, but instead used clustering metrics to evaluate.

The organizers also provided a standardized mapping system for mapped evaluation, which made use of each system's labels on one portion of the data (the "training" portion) to derive the most likely mapping to the gold standard labels, and then used that mapping to calculate the system's F-Score on the rest of the data (the "test" portion)[7]. The mapping matrix M is defined as follows:

$$M_{i,j} = P(S_i|C_j) = \frac{|S_i \cap C_j|}{|C_j|}$$

In other words, each cell $< i, j >$ in the matrix contains the proportion of times where

[6]The authors refer to these as the 'unsupervised' and 'supervised' evaluation methods, but we will use 'cluster-based' and 'mapped' to avoid confusion

[7]It is important to note that the labels of the "training" portion are not used in any way for actual training of the model, since the entire system is unsupervised. They are only used to provide a mapping to the gold standard, for evaluation purposes. The "training" part could more accurately be called the *mapping* portion.

an instance of the mapping data assigned to cluster C_j had the gold-standard sense-tag S_i.

Then, given a cluster assignment vector $v_x = (p_1, p_2, ..., p_m)$ produced by the WSD system for each instance x, where p_j is the probability of assigning that instance to cluster C_j, the mapped sense assignment scores are calculated by multiplying the assignment vector with the mapping matrix. The final mapped (gold-standard) sense assignment for that instance is chosen by selecting the sense with the highest mapped assignment score.

$$Assignment(x) = \underset{i}{\text{argmax}}\ v_x \cdot M$$

Under the cluster-based evaluation setting, the one-cluster-per-word baseline outperformed all the systems except one, which was only marginally better. It is important to keep in mind that labeling all instances with a single sense does not truly comprise a feasible baseline system. The cluster-based evaluation ignores the actual labelling, and due to the dominance of the first sense in the data, encourages a single-sense approach. In addition, as stated above, the evaluation was done using coarse-grain OntoNotes senses, which further amplified the predominant-sense problem. For the purposes of this work, therefore, we focused on the mapped evaluation.

The best performing system in the mapped evaluation setting was I2R (Niu et al., 2007) described in Section 5.2. Under this setting, most of the participating systems outperformed the most-frequent-sense (MFS) baseline, and those that didn't obtained only slightly lower scores.

5.4.4 Sense Induction Procedure

Sense induction methods do not use labeled data in any part of the process. However, they do make use of large amounts of unlabeled data, in order to get as much information as possible about the characteristics of the data. While this unlabeled data is sometimes also called "training data", we will refer to it as "learning data", to differentiate it from labeled training data used in supervised systems.

In order to induce the senses of a target word, we created a combined dataset consisting of all the instances of the word extracted from the large learning corpus, together with the instances extracted from the much smaller test data. We then ran the Gibbs inference procedure on the combined dataset. Due to the difference in size (three orders

of magnitude) between the learning corpus and the test set, the sense-cluster distinctions are almost entirely influenced by the properties of the former. The output of the process is sense-cluster assignment probabilities for every instance in the combined dataset, but for evaluation purposes we are only interested in the assignments of the test set instances.

5.5 Sense Induction Using Layered LDA

Before presenting the quantitative results of our experiments, we give an example of the sense-clustering produced by our system in Section 5.5.1. Our experiments address several issues involved with using our layered LDA model for sense induction. The first issue is that of model selection. Our model and the induction framework contain several parameters that can be adjusted to better model the data. We examine the effects of these parameters on system performance in Section 5.5.2. Another important issue is the selection of information sources (layers) used by our model. We address this in the experiments in Section 5.5.3. We also examine the issue of cross-domain learning. As mentioned in the introduction to this chapter, sense induction frees the system from dependence on a fixed sense inventory, thereby enabling use on new tasks and languages. However, there is still an implicit dependence remaining. Sense induction methods typically rely on large (unlabeled) corpora for learning. These are often standard, publically available, machine readable corpora, not necessarily in the domain of interest. This leads to the question: in this framework, what are the effects of cross-domain learning? More specifically, how effective is learning from a general corpus? Is it better to train on a small in-domain corpus, or a large out-of-domain one? Do we have to tune model parameters separately for each domain? How detrimental is cross-domain learning, and what can we do to minimize negative effects? To answer these questions, in Section 5.5.4 we compare a system which learns from an out-of-domain corpus (BNC), to our main system, which learns from an in-domain corpus (WSJ). Finally, in Section 5.5.5, we compare the performance of our methods to state-of-the-art.

5.5.1 Example of System Output

The OntoNotes sense definitions and automatically induced clusters for the words *drug* and *power* are presented in Tables 5.3 and 5.4, respectively. The senses were

"Production"	**"World Politics"**	**"Financial"**	**"National Politics"**
plant	party	plant	bank
company	government	co.	president
computer	political	nuclear	congress
nuclear	military	million	state
electric	president	unit	government
system	economic	utility	security
year	U.S.	electric	federal
U.S.	people	company	executive
utility	world	light	company
price	soviet	corp.	court
line	country	power	law
market	struggle	share	veto
industry	election	inc.	authority

OntoNotes Sense Definitions for *power*:

- **Sense 1** An ability to control or influence.
- **Sense 2** Entity that possesses ability to control or influence.
- **Sense 3** Exerted physical force.
- **Sense 4** A mathematical measurement.

Table 5.3: Manual sense definitions and induced sense-clusters for the word *power* extracted from the WSJ using a single $\pm$10-word layer. Cluster labels were manually assigned.

"Enforcement"	**"Treatment"**	**"Industry"**	**"Research"**
U.S.	patient	company	administration
administration	people	million	food
federal	problem	sale	company
against	doctor	maker	approval
war	company	stock	FDA
dealer	abuse	inc.	patient
government	aid	market	test
official	user	product	market
enforcement	test	co.	U.S.
testing	prescription	U.S.	approve
charge	cost	sterling	treat
trafficker	year	prescription	aid
money	alcohol	drug	study
president	effect	generic	product
abuse	addict	analyst	treatment
program	treatment	industry	develop
law	Dr.	pharmaceutical	receive

OntoNotes Sense Definitions for *drug*:

- **Sense 1** Medicines. A substance that affects the body in some legal, usually-beneficial way. Does not apply to narcotics.
- **Sense 2** Narcotics. A substance, usually illegal, that causes bodily pleasure or some other reaction. Has a very negative connotation.

Table 5.4: Manual sense definitions and induced sense-clusters for the word *drug* extracted from the WSJ using a single $\pm$10-word layer. Cluster labels were manually assigned.

induced using the in-domain learning corpus (WSJ) and a single layer consisting of words occurring in a $\pm$10-word window. The lists contain the most likely content words to appear in the window for each sense-cluster. As we can see, the induced sense distinctions only roughly correspond to those in the lexicon. For *drug*, for instance, the first two induced senses match the first OntoNotes sense, whereas the third and fourth sense-clusters correspond to the second. For *power*, the second and fourth OntoNotes senses are missing. In fact, this is in keeping with the manual sense labelling in the Semeval test data, where the fourth sense is completely absent, and the second sense is very infrequent (approximately 6%). This is a good example of a case where a fixed lexicon is unsuitable for the specific domain at hand, but automatically induced senses accurately match the data. Note that several words are shared between two, or more of the clusters. However, some of these have different shades of meaning in the different contexts (e.g., the word *treatment* in the second and fourth sense-clusters in Table 5.4).

5.5.2 Model Selection

There are several parameters that need to be addressed when using our model. The question of the optimal number of clusters in an unsupervised clustering problem is an important and difficult one. In our case, this means deciding on the desirable number of sense clusters (see Section 5.2 for treatment of this issue in previous work). Additionally, our system contains several hyperparameters which can be adjusted to better model the data. The α and β hyper-parameters determine the sense-cluster and feature distributions, respectively. Another parameter regards the decision of when the model has converged. Finally, our layered model adds the option of different weights for each of the layers.

In our experiments we examined the two highest level parameters, namely, the number of sense-clusters and value of α. The rest of the parameters were set to common default values. The β parameter was set to 0.1 (in all layers). This value is often considered optimal in LDA-related models (Griffiths and Steyvers, 2002). The number of convergence iterations was set to 2000. For simplicity, we chose uniform weights for the layers. Due to the randomized nature of the Gibbs inference procedure, all the results reported here and in the following sections are average scores over ten runs.

In order to determine the best value for the α parameter, we used the Senseval 2 lexical sample dataset for tuning, and experimented with values ranging from 0.005 to 1. Based on the results of this experiment, shown in Figure 5.7, we set the value

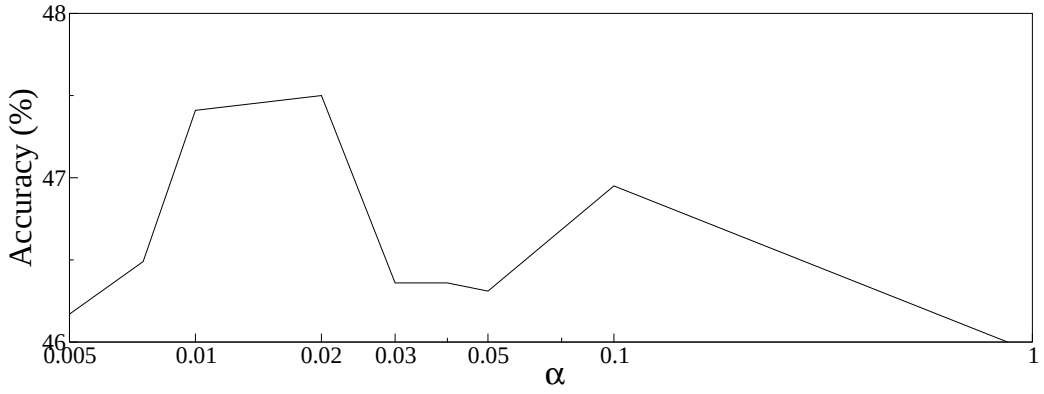

Figure 5.7: Model performance with varying values of the α parameter on the Senseval 2 dataset.

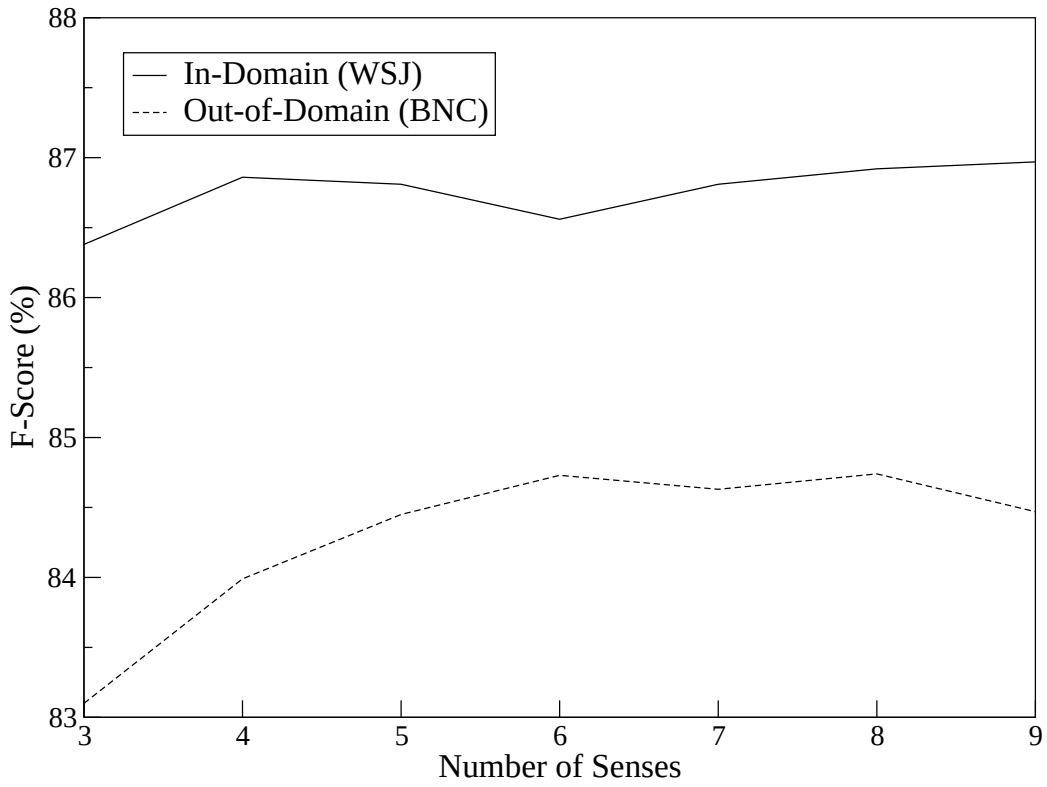

Figure 5.8: Model performance with varying number of senses on in and out-of-domain corpora with a single ± 10-word layer.

$\alpha = 0.01$	$\alpha = 0.02$	$\alpha = 0.03$
86.7	86.9%	86.9%

Table 5.5: Results (F-Score) of model selection experiments for the in-domain (WSJ) system.

Single Layer		Remove One			Combination	
Layer	F-Score	Layer	Diff.	F-Score	Layer	F-Score
10w	**86.9%**	-10w	-0.2%	83.1%	**10w+5w**	**87.3%**
5w	86.8%	-5w	-0.3%	83.0%	5w+pg	83.9%
1w	84.6%	-1w	-0.3%	83.0%	1w+ng	83.2%
ng	83.6%	-ng	-0.3%	83.0%	10w+pg	83.3%
pg	82.5%	-pg	-0.6%	82.7%	1w+pg	84.5%
dep	82.2%	-dep	**+1.4%**	84.7%	10w+pg+dep	82.2%
MFS	80.9%	all	–	83.3%	MFS	80.9%

Table 5.6: Results (in-domain) for single layer (left), leave-one-out (center) and selected combination (right) experiments. Also shown are the most-frequent-sense (MFS) baseline, and results of the combined system, containing all layers (all).

of α to 0.02 in all our subsequent experiments. In order to examine the effectiveness of using an external tuning dataset to determine the parameter values, we performed additional experiments using the Semeval data (our test set). We explored the effect of small changes from the optimal value of *alpha* (as determined by our tuning experiment). The results of these experiments are shown in Table 5.5. The effects are quite minor ($\leq$0.2%). This indicates that there is little need to fine-tune the α parameter, and relying on a value obtained using an external tuning set (as we did) is sufficient.

To address the issue of model-order, we experimented with values ranging from three to nine sense-clusters. Figure 5.8 (solid line) shows the results obtained for different numbers of sense-clusters for the in-domain system (WSJ), using the a single $\pm$10 word layer. For this system, performance peaks at four sense-clusters, which is a reasonable result, given that this is close to the average ambiguity in the test data (see Table 5.2).

5.5.3 Layer Analysis

Table 5.6 presents the results of a series of experiments designed to investigate the relative contributions of the individual layers to the combined (layered) model. All experiments employed the optimal parameters determined in the previous section ($\alpha = 0.02$, with 4 sense-clusters).

The first set of experiments compares the performance on the induction task using each of the individual layer on their own. This corresponds to running a basic (non-layered) version of our LDA model (as in Figure 5.4), using only a single information source as input. The results are shown on the left side of the table. The layer composed of words co-occurring within a $\pm$10-word window, and representing wider, topical, information gives the highest scores on its own. It is followed by the $\pm$5-word and collocation (1w) windows, which represent more immediate, local context. The word n-grams and the part-of-speech n-grams, on their own, achieve lower scores, largely due to sparseness. The lowest-scoring single layer is the dependency layer, which gives results that are only slightly above the most-frequent-sense (MFS) baseline. Dependency features are very specific (containing information about the type of dependency, whether the target word is the head or the dependent, and the other word involved in the relation). This means that it is very informative when present, but extremely sparse. All the systems outperform the most-frequent-sense baseline.

The center portion of the table shows the results obtained when running the layered model with all but one of the layers as input. We can use this information to determine the contribution of each layer by comparing (middle column) to the combined model with all layers (all). Because we are dealing with multiple layers, there is an element of overlap involved. Therefore, each of the word-window layers, despite relatively high informativeness on its own, does not cause as much damage when it is absent, since the other two layers compensate for the topical and local information. The absence of the word n-gram layer, which provides specific local information, does not make a great impact when the collocation layer and the part-of-speech n-gram layer are present. Finally, we can see that the extremely sparse dependency layer is determental to the multi-layer model as a whole, and its removal *increases* performance. The sparsity of this layer means that there is often little data on which to base a decision. In these cases, the layer contributes a close-to-uniform estimation of the sense distribution, which confuses the combined model.

On the right side of the table we present the results for a selected set of layer combi-

$\alpha = 0.01$	$\alpha = 0.02$	$\alpha = 0.03$
84.1%	84.7%	84.4%

Table 5.7: Results of parameter-variation experiments out-of-domain (BNC)

nation experiments. The benefits of combining information sources are evident in one case. The 10w+5w combined system produces the best performance, outperforming each of its individual component layers by 0.5%. Other pairwise combinations result in scores which are inbetween the scores of their components, and do not perform as well as the best individual layer in the pair. The lesson we can learn from these results is that the layered model can provide improved results by combining multiple sources of information, but these must be carefully selected. Combining two layers with similar performance on their own results in improved scores. However, if one of the components is considerablly weaker than the other, it will tend to effect the combined system, resulting in a lower score than that of the strong component by itself. The results presented here also address the issue of the naive independence assumption underlying our layered model. Using a combination of layers which are more independent of one another, as in the 10w+pg+dep combined system, is not sufficient to improve results, and improvements can be gained even when the independence assumption is strongly violated, as in the case of the 10w+5w combination. These results suggest that it is the relative strength of the component layers, rather than their mutual independence, which affects the performance of the combined system.

5.5.4 Cross-Domain Learning

5.5.4.1 Model Selection

Figure 5.8 (dashed line) shows the results obtained for different numbers of sense-clusters in the out-of-domain (BNC) system, using a single $\pm$10-word layer. For this system, the best results were obtained using twice as many sense-clusters as required by the in-domain system (solid line in the figure). This can be attributed to the loss of accuracy resulting from the shift in domain. The coarse sense-divisions of the learning domain do not match those of the target domain (as seen in the example below). Instead, finer granularity is required in order to encompass all the relevant distinctions. Table 5.7 presents the results of our experiments with small variations of α on the Semeval data. Once again, the differences are relatively small, and the optimal value for

the tuning dataset (0.02) gives the best performance on our test data, as well. These model-selection experiments confirm the conclusion reached in Section 5.5.2 that an external tuning set provides a reasonable estimate of the α parameter, which does not vary greatly across domains. On the other hand, the correct selection of the number of sense-clusters has a bigger effect (a 0.75% increase between 4 and 8 sense-clusters in this case, and up to 2% in other settings with which we experimented).

Table 5.8 presents the automatically induced clusters for the word *drug*, this time using the out-of-domain corpus (BNC) for the induction process, and eight topics, instead of the four used with the in-domain corpus. While there is some correspondence with the sense-clusters in Table 5.4 (as indicated by the assigned cluster labels), the differences between the two corpora are clearly represented. The WSJ focuses on the financial aspect, while the BNC is directed towards a wider, general interest, audience.

5.5.4.2 Layer Analysis

Table 5.9 presents the results of a series of experiments, similar to the ones in presented in Section 5.5.3, this time using an out-of-domain system, trained on the BNC. The general trends for individual layers (left) and all-but-one layer systems (center) are similar, although some interesting differences are apparent. The sparser layers, notably word n-grams and dependencies, fare comparatively worse. This is expected, since the more precise, local, information is likely to vary strongly across domains. Even when both domains refer to the same sense of a word, it may to be used in a different immediate context, and local contextual information learned in one domain will be less effective in the other.

Another observable difference is that the combined model excluding only the dependency layer does better than each of the single layers. Due to the discrepancies between domains, each individual layer is less effective, and the benefit of combining as much data as possible outweights the negative influence of the weaker layers. In fact, the all-layers out-of-domain system outperforms the all-layers in-domain one (compare bottom-center cell in both tables).

Looking at the results of our layer combination experiments (right portion of the table), we see that the conclusions we drew from the combination experiments with the in-domain corpus (see Section 5.5.3) hold true here, as well. The combined systems which outperform their individual components are 10w+5w (producing the best results of any of the out-of-domain systems), and 1w+pg. In each of these systems, the component layers have similar performance on their own.

"Trafficking"	**"Wonder Drug"**	**"Abuse"**	**"Enforcement"**
trafficking	think	effect	police
against	people	alcohol	charge
trafficker	addict	disease	court
traffic	involve	treatment	test
U.S.	life	cause	supply
cartel	drink	pain	drug
government	help	addiction	jail
charge	effect	take	dealer
enforcement	mean	patient	use
state	feel	addictive	arrest
control	wonder	chemical	possession

"Treatment"	**"New Meds"**	**"Issue"**	**"Research/Industry"**
alcohol	patient	drink	company
problem	treatment	addict	food
health	intravenous	abuse	research
abuse	effect	crime	administration
drug	therapy	addiction	world
people	HIV	life	U.S.
service	ulcer	sex	price
patient	anti-inflammatory	centre	product
prescribe	non-steroidal	alcohol	market
treatment	concentration	family	cost
help	report	violence	pharmaceutical

Table 5.8: Induced sense-clusters for the word *drug* extracted from the BNC using a single ± 10-word layer. Cluster labels were manually assigned.

Single Layer		Remove One			Combination	
Layer	F-Score	Layer	Diff.	F-Score	Layer	F-Score
10w	84.6%	10w	-0.8%	83.3%	**10w+5w**	**85.5%**
5w	84.6%	5w	-1.3%	82.8%	5w+pg	83.5%
1w	83.6%	1w	-0.6%	83.5%	1w+ng	83.5%
pg	83.1%	pg	-0.9%	83.2%	10w+pg	83.4%
ng	82.8%	ng	-1.2%	82.9%	1w+pg	84.1%
dep	81.1%	dep	**+0.6%**	**84.7%**	10w+pg+dep	81.7%
MFS	80.9%	all	–	84.1%	MFS	80.9%

Table 5.9: Out-of-domain results for single layer (left), leave-one-out (center) and selected combination (right) experiments. Also shown are the most-frequent-sense (MFS) baseline, and results of the combined system, containing all layers (all).

5.5.4.3 Corpus Size vs. Domain

Figure 5.9 shows the scores achieved by the model using increasingly large portions of the corpora. The system uses a single ± 10 word layer, and the parameters are those determined as optimal in the previous sections ($\alpha = 0.02$, four and eight sense-clusters for the in- and out-of-domain systems, respectively). In general, for the in-domain system, the increase in data seems to improve accuracy, but the differences are small, and are sometimes overweighted by the randomness of the sampling algorithm. For the out-of-domain setting, increasing the corpus size does not show a consistent benefit. From these results it is clear that using a small amount of in-domain data is preferable to using a very large out-of-domain corpus. Even using the entire out-of-domain corpus results in lower scores than those achieved with 10% of the in-domain one.

5.5.5 Comparison to State-of-the-Art

Table 5.10 compares the results of our in- and out-of-domain layered-LDA systems to the the top two systems in the Semeval induction task. Both LDA systems significantly outperform the most-frequent-sense (MFS) baseline ($p < 0.01$ using a χ^2 test). Our best in-domain system outperforms the highest-scoring system in Semeval (I2R), while our out-of-domain system outperforms the second-best system (UMND2), although the differences are not statistically significant. The difference between our in-domain and out-of-domain systems is significant (at $p < 0.01$).

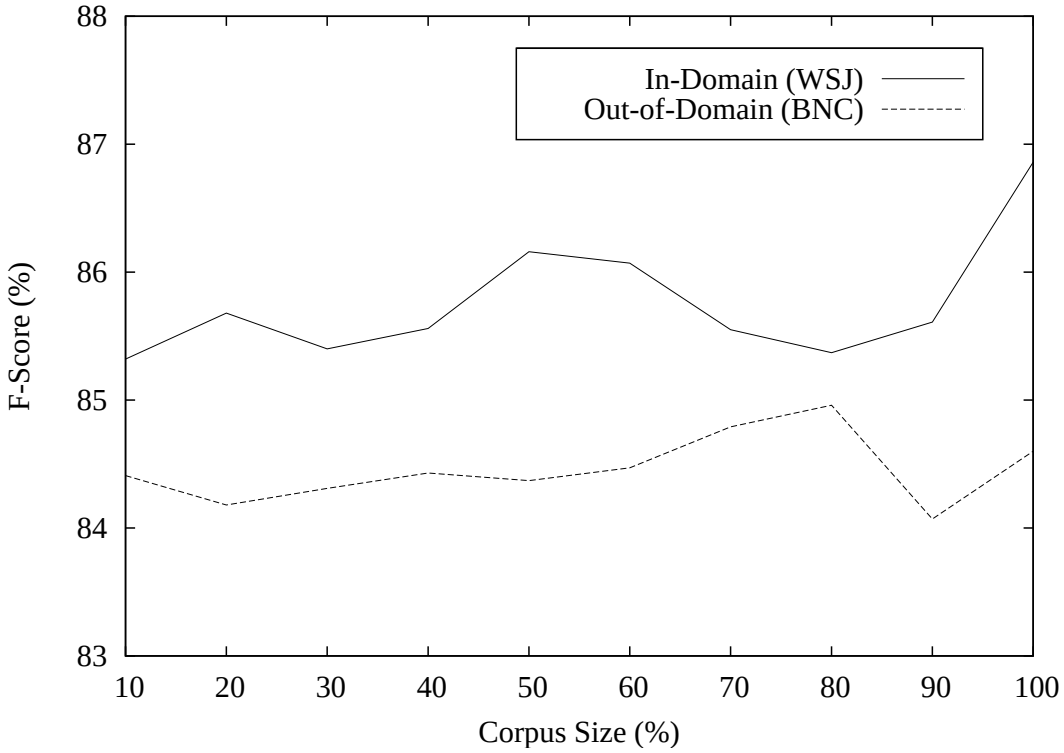

Figure 5.9: Model performance with increasing sizes of in-domain (solid) and out-of-domain (dashed) learning data.

System	F-Score
LDA-WSJ (10w+5w)	**87.3%**
I2R	86.8%
LDA-BNC (10w+5w)	**85.5%**
UMND2	84.5%
MFS	80.9%

Table 5.10: F-score of best-performing Semeval-07 systems and our LDA models on the sense induction task, using mapped evaluation.

5.6 Discussion

Our experiments were designed to investigate several aspects of our layered LDA model. In this section, we will briefly discuss our experimental results and their implications.

Layers vs. Ensembles The ensemble methods described in Chapter 3 showed us the benefit of combining the opinions of several systems. However, the individual systems in the ensemble are unaware of each other (with the possible exception of the Arbitor ensemble, see Section 3.4.2), and the benefits of the combined opinions come only in a post processing stage. In the layered model, on the other hand, the combined sources of information are present from the input stage, and are expected to influence each other during the learning stage itself, thus improving performance.

Model Selection The first aspect was that of model selection. We looked at the effects of the values of high-level parameters and determining the correct number of clusters. Our experiments indicate that our model is relatively robust to small adjustments of the α parameter, which influences the cluster distribution. This means that we can use an external tuning set to estimate the best parameter values with little risk of impairing our model. The issue of model order, i.e., determining the correct number of clusters, is a more complex one, and has a bigger effect on performance. Although using a number of clusters corresponding to the average ambiguity provided good performance in our experiments, it would be preferable to determine this parameter in a more principled manner, preferably on a word-for-word basis.

Layer Analysis We explored the potential of our extention to the LDA model, which allows it to make use of multiple layers of information. Our experiments compared a variety of feature categories, designed to capture local, topical and syntactic informantion, and the interactions between them in the layered model. The results show that combining different information sources can be beneficial, and lead to performance gains, but the layers must be carefully selected. Combining layers which differ strongly in individual performance can lead to decrease in performance. On the other hand, layers with similar individual performance can improve results, even when the independence assumption inherent in the model is violated. There is still much scope for exploring the layered aspect of the model, including a closer look at how independence effects performance, and whether differential weighing of the layers can provide

a way to exploit even the weaker information sources.

Co-occurrence vs. Syntax A disappointing observation arising from our results is that our best performance was achieved using only co-occurrence information, while making use of syntactic dependency information actually hindered the model. Since we are dealing with a token-based approach, where each instance of a word in treated individually, dependency data is very sparse. Most instances participate in only one or two dependency relations, and this has a strong detrimental effect on performance. In the previous chapter, on the other hand, we showed that dependency information can improve accuracy over simple co-occurrence information (see Table 4.3) when dealing with word *types*, rather than tokens. It may also be the case that dependency information would be more useful for the treatment of verbs, rather than nouns. Verbs commonly participate in more dependency relations, and syntactic information can be expected to play a larger part in their disambiguation.

Domain Independence in Sense Induction Our experiments addressed the issue of cross domain learning and its effects on the performance of the model. From the results, it is clear that learning from an in-domain corpora provides much better performance than using an out-of-domain one. Even the help of a much larger learning corpus and the combination of multiple sources of information are not enough to compensate for the cross-domain effect. This serves to emphasize the advantages of a completely unsupervised method, which requires only plain text as input, and can therefore be applied easily to the domain at hand. As demonstrated in the example in Table 5.3, the automatic induction of senses can help model the target data more accurately than relying on an external fixed sense inventory. Both these characteristics of our method (freedom from annotation and independence from a pre-specified dictionary) make it especially suited for wide coverage WSD, and applicable to many nature-language tasks.

5.6.1 Summary

We presented an unsupervised, generative, probabilistic method for sense induction. This provides a solution to the problem of dependence on a fixed sense-inventory, which severely limits the potential uses of unsupervised WSD methods. Our method has many of the advantages common in a supervised framework, but does not require labeled data. The system is based on an extension of the Latent Dirichlet Allocation

model, which we referred to as Layered-LDA. Our extension is general, and can be used for any task where several sources of information are available. Our method achieves state-of-the-art results on the unsupervised sense induction task.

Chapter 6

Conclusions and Future Directions

> Mellvile, as a great author, used one word to convey two ideas, as opposed to the typical scientific paper, which can go for pages without conveying any ideas at all.
>
> Michael Lesk

This chapter concludes our thesis. In Section 6.1, we summarize the main findings and contributions of the thesis. Section 6.2 addresses the issue of application, i.e., the use of WSD methods as part of a larger system designed to perform a real-world task. We conclude in Section 6.3 with a discussion of future research directions.

6.1 Findings

The work in this thesis addresses the performance gap separating unsupervised and supervised WSD. It takes a deep look into the nature of this gap, explores its causes, and presents solutions to help bring unsupervised methods closer to the level of performance common to supervised ones. We addressed classic unsupervised methods which make use of a dictionary (see Section 2.1.1) in Chapter 3. In Chapter 4 we presented ways to automatically sense-label training data, thus enabling a supervised methodology without manual annotation. We addressed the completely unsupervised task of sense induction in Chapter 5. Our research also provides some important insights regarding the relative strengths and weaknesses of supervised and unsupervised methods in computational linguistics in general. The main findings of the thesis are presented in detail below.

A standardized framework for comparison and analysis of unsupervised WSD methods. We designed a framework which provides all the necessary infrastructure to allow the comparison, evaluation, and detailed analysis of existing unsupervised methods on the same data, under uniform conditions. To our knowledge, such a setting was not available previously, nor was such a comparison performed in the past. Our experiments compared the performance of four unsupervised WSD algorithms, employing different representations, approaches and methodology (context overlap, lexical-chains, structural-semantic interconnections, and predominant-sense detection using similarity metrics). We also examined the utility of using the most frequent (first) sense estimated by each of the algorithms to label all instances of the data. The results of our experiments lead to the following conclusions. Type-based disambiguation (i.e., using the most-frequent-sense for all instances) outperforms token-based, context-specific, disambiguation. Although labeling all instances with a single sense does not provised an applicable WSD solution on its own, the first sense can provide a reliable fallback for token-based methods in cases where the context is not sufficiently informative. Our examination of the different approaches used in previous algorithms brought to light the effectiveness of using distributional and semantic similarity metrics, which we employed in our unsupervised data-creation method. The detailed comparison of the accuracy of the different methods demonstrated their complimentary nature. Each method performed well on a different group of words, with little overlap. This finding led to the subsequent development of unsupervised ensemble methods.

Ensemble combinations. We developed unsupervised ensembles for improving the performance of a group of WSD methods. We examined several ensembles (arbiter, probability-based, voting, and rank-based) designed to operate on the basis of sense-labeled output, with no assumptions regarding the methodology employed by the component algorithms. We found that even simple ensembles achieve better results than individual components and outperform state-of-the-art algorithms on standard evaluation sets. Our experiments show that the best performing ensemble is the rank-based one. This ensemble considers information regarding all the senses, under a weighting scheme that is independent of the algorithm's underlying methodology. The ensembles serve an important function in demonstrating a way in which the differences in formulation and approach to the disambiguation problem, that had previously been a hindrance to the development of accurate WSD system, can be harnessed to improve performance. The ensemble methods are very helpful in cases where a WSD sys-

tem already exists, and the cost of developing a more accurate system from scratch is prohibitive. In these circumstances, they can be used to improve performance with very little additional effort. Another important benefit of our ensemble methods is to provide a strong fallback option in cases where context is not sufficiently informative.

Finally, our work on unsupervised ensemble methods also represents an important contribution to unsupervised learning in general, since it demonstrates the benefits that can be gained from employing simple ideas drawn from supervised methodology in an unsupervised framework.

Unsupervised creation of labeled data. We described an unsupervised method which uses distributional and semantic similarity metrics to automatically sense-label training data, thus enabling the use of supervised classifiers in an unsupervised framework, and reducing the gap between the two settings. We made use of distributional similarity to detect words with similar context-statistics in the corpus (distributional neighbors), and employed semantic similarity metrics to associate them with senses of the target word. We then extracted instances of the distributional neighbors, along with their context, from the corpus, replacing each neighbor with the target word and labeling it with the associated sense. This procedure produced our sense labeled dataset. We trained three supervised classifiers (SVM, maximum-entropy, and label propagation), based on different learning approaches, on our automatically created data, and on data created using previous methods proposed in the literature. We evaluated on two standard datasets, and compared to standard unsupervised WSD methods and to the use of manually labeled data, as an upper bound. Our results showed that classifiers trained on our automatically created data can surpass the performance achieved by previous methods of automatic data-creation and outperform state-of-the-art unsupervised methods. We further showed that coverage of secondary senses varies between classifiers. Using an SVM as the classifier resulted in coverage approaching that of using manual data, making it the preferred choice in cases where secondary senses are especially important.

We found that using co-occurrence-based distributional neighbors rather than dependency-based ones resulted in only a small decrease in performance ($0-4.5\%$), and was still preferable to using other data-creation methods. This means that our approach can be used in cases where accurate syntactic parsers are unavailable. We also found that data created with our method is similar to manually labeled data in terms of sensitivity to the coarseness of the sense distinctions. Our approach can therefore be used

for tasks requiring either fine or coarse sense granularity. Comparison of the different classifiers revealed that the label-propagation classifier had the smallest decrease in performance when using our dataset instead of the manually labeled one. We attributed this to the fact that this classifier takes into account the character of the test set, as well as that of the training data. This allowed it to compensate for the differences between our pseudo-instances and real instances of the target word, as found in the test data. Our method was successful for a variety of classifiers. This implies that improvements to supervised methods (through the development of better classifiers) can be easily transferred to the unsupervised setting. Our approach therefore represents a significant step in bridging the performance gap between unsupervised and supervised methods.

From the point of view of unsupervised learning, our method proves the effectiveness of unsupervised data creation methods, and opens the way for a similar methodology in other tasks which employ machine learning algorithms trained on labeled data, such as parsing and relation extraction.

A Bayesian model for sense induction. We presented a sense induction system based on a probabilistic generative model, which is independent of a fixed sense inventory. We introduced a novel point of view of the sense induction task. Whereas previously the task has been treated as a standard clustering problem, we view it from a more language-oriented perspective. Our approach postulates that the observed data (context surrounding the ambiguous word) is generated with the intent of communicating the latent meaning of the word. Our model provides a principled way to incorporate a wide range of informative features in the induction process. We adapted the LDA model originally designed for modelling text generation to the task of sense induction. We extended the model to allow the use of multiple sources of information. We investigated the properties of our model, including the effects of model parameters, the selection of information sources, and cross domain learning. We compared the results of our model to those of state-of-the-art sense induction methods based on clustering.

We found that appropriate selection of model order has greater effect on performance than variation in model parameters. In addition, an external tuning dataset can be used to reliably estimate the desired value of the parameters. In depth analysis of several feature categories and their contribution to the combined model revealed that performance can benefit from the combination of several information sources, pro-

vided they are strong predictors on their own. This is necessary to prevent weaker components from having a negative effect on the model as a whole. On a standard evaluation dataset, our model outperforms state-of-the-art methods for unsupervised sense induction. The induced senses match the distinctions present in the data. Our model is therefore suitable for a variety of tasks and domains, where methods based on a fixed sense-inventory (both classical unsupervised and supervised WSD methods, see Section 2.1.1) may suffer from the noise introduced through irrelevant senses or unnecessary distinctions.

Finally, our model represents a general extension of LDA, designed to make use of multiple sources of information. It can therefore be employed on a variety of tasks, other than WSD, where such information exists.

To summarize, in this thesis we have explored the nature of the performance gap separating unsupervised and supervised WSD. We have addressed many of the fundamental issues contributing to this gap, and presented our solutions to these problems. As a first step, the incompatibility, diversity, and lack of standardization of existing unsupervised WSD systems were addressed using ensemble methods. Next, we addressed the problematic aspect of problem formulation and difference in approaches that characterizes unsupervised WSD. We presented an unsupervised process for creating sense-labeled training data, which retains the freedom from manual annotation, while transferring the actual disambiguation to the hands of the more accurate and powerful supervised methods. Finally, we turned our attention to the restrictions imposed by a predefined sense inventory. We presented a system for unsupervised sense induction, which allows unsupervised WSD to be easily integrated into natural-language applications, and tailored to a specific task and domain, without the need to define a new purpose-built sense inventory and corresponding training dataset. Our research also provides some important insights regarding the nature of the performance gap separating supervised and unsupervised methods in WSD and in computational linguistics in general. All our methods surpass current state-of-the-art performance on their respective unsupervised tasks, and represent a significant step in closing the gap in WSD.

6.2 Applications

An important aspect not addressed in this thesis, but which requires mention, is the issue of application. Though the stand-alone WSD setting is often required in order

to remove external influences and focus purely on the disambiguation problem, it is of little practical use in and of itself. Integration into a real-world application is important both as a realistic evaluation of performance and in order for any WSD system to have pragmatic value. The methods and systems we presented in this work each have specific characteristics making them suitable for different uses and applications.

The ensemble methods described in Chapter 3 were designed primarily to help improve the performance of existing WSD systems. These methods are of potential use for legacy systems, which contain an existing WSD component. Our ensembles present a quick and easy way of improving performance without the cost of developing a new WSD component from scratch and ensuring its compatibility with system requirements. The existing component can be integrated into an ensemble along with other available WSD algorithms. Our experiments show that results of a state-of-the-art system can be improved even when the ensemble contains relatively weak members. Since our ensemble methods operate on the basis of the predominant sense methodology, disregarding context, they can be used in combination with supervised WSD methods that take context into account. For instance, the ensembles could provide a strong fallback option for supervised methods when encountering unseen words or words with uninformative contexts.

The data-creation method presented in Chapter 4 represents a highly-versatile tool for disambiguation. It can be employed to replace or enhance (as described in Section 6.3, below) any existing supervised WSD component. It is thus ideally suited for easy integration into natural-language applications which have previously relied on supervised methods for accuracy, and been restricted by their limitations. Use of this method allows the expansion of such applications to encompass new domains where sufficient training data is unavailable.

Another potential application for both our ensemble methods and for classifiers trained on our automatically-labeled data, would be to create preliminary annotations, under the "annotate automatically, correct manually" methodology. This methodology can be used to reduce manual effort and provide high volume annotation, as demonstrated in the Penn Treebank project.

Sense induction holds great promise in terms of application, since it learns directly from the data, and the induced sense distinctions are therefore those which are relevant to the task and domain at hand. As mentioned in Chapter 5, recent work in machine translation (Vickrey et al., 2005) and information retrieval (Véronis, 2004) indicates that induced senses can lead to improved performance in places where methods em-

ploying a fixed dictionary have previously failed (Carpuat and Wu, 2005; Voorhees, 1993). Aside from the benefits of induced senses, rather than fixed ones, our sense-induction model is especially suited for use as a component in an application due to its probabilistic nature. The probabilistic formulation allows for easy integration with other components through mixture and product models. Yet another advantage of our model is the easy integration of additional sources of information, which can be specific to the task at hand. For instance, we might want to include contextual information from the target language as well as the source, if using the model as part of a translation system, or include relevant meta-data if the system is being used for information retrieval.

6.3 Directions for Future Work

The work in this thesis opens many avenues for further research. The unsupervised ensemble methods presented in Chapter 3 can be extended in several directions. The ensembles we explored were based on simple methodologies, designed to impose as few restrictions as possible on the component systems. More sophisticated ensemble methods, which have more knowledge on which to base their decisions, could give better performance. For instance, taking into account the algorithms' confidence in their classification, the ensemble could choose to ignore members with low-confidence, on a per-instance basis. In addition, it could choose when to make use of the context-based classification provided by the ensemble members, and when to default to the document-wide predominant sense. Another direction to pursue is integrating more members into the ensembles. This has the potential for increasing their accuracy and robustness. Possible additions include not only domain driven disambiguation algorithms (Strapparava et al., 2004) but also graph theoretic ones (Mihalcea, 2005b), as well as algorithms that quantify the degree of association between senses and their co-occurring contexts (Mohammad and Hirst, 2006). Increasing the number of components would also allow more sophisticated combination methods such as unsupervised rank aggregation algorithms (Tan and Jin, 2004).

In Chapter 4 we introduced an unsupervised method for creation of labeled training data. This method, too, presents many possibilities for further research. Providing an unsupervised method which differs from supervised ones only in its training data makes it very easy to integrate the two methodologies. This could be done in several ways. One option is to explore ways to merge manually and automatically

labeled data. Such a combination could be used to inflate a small manually-labeled dataset. The integration would require a way of strengthening the importance of the few manual labels so they are not overwhelmed by the automatic ones, and needs to take into account the different nature of the two components (actual instances of the target in the manually labeled data versus pseudo-instances, created from distributional neighbors in the automatically-created portion). This presents an interesting challenge from a research perspective. Another option is to perform the integration on a per-word basis. Under this setting, our unsupervised system can be used for most cases. However, for words for which there is already sufficient training data, or when our unsupervised methodology does not provide the desired accuracy, supervised methodology can be employed. This greatly reduces the burden of manual annotation to a few specific cases. It also makes it relatively easy to shift domain, since whenever new words or senses are encountered (e.g., terminology or senses that are specific to the new domain), our method can be used to provide the missing information. A further possibility is to shift the focus of the manual labor from the task of labeling examples to that of selecting informative neighbors for senses of ambiguous words. This could represent an enormous reduction in the amount of manual labour required for producing a training dataset, since a single informative neighbor can provide a large number of training instances. Finally, the method we presented highlights the effectiveness of automatic data-creation as an unsupervised methodology. It would be very interesting to see whether this methodology could be successfully employed in other tasks where labeled training data is used, such as parsing or relation extraction.

The layered-LDA model for sense-induction presented in Chapter 5 suggests many interesting research possibilities. Our experiments used a set of layers composed of a simple agglomeration of features commonly used in the field and shown to be informative for WSD. However, they may not be the best choice for integrating in a model such as our own, which assumes independence between the layers and prefers layers with similar individual performance (as shown in our experiments). Further study into the optimal choice of information sources to use as layers could be beneficial to our sense-induction system. The model itself contains several elements which could benefit from further study. One issue is the weighting of the layers. Our model provides the option of assigning different weights to the different information layers. Our experiments show that some of the layers we used were much more accurate and informative than others. However, in our system, we chose to weight all the layers equally for the sake of simplicity. Determining the optimal layer weights presents an interesting

problem with strong potential for improving the performance of the model. Another issue is the correct tuning of model parameters. In general, we chose not to focus on the issue of parameter estimation in our work and therefore did not include a comprehensive study of its influence on the model. All model parameters were set either to standard default values, or estimated on a separate, held out, dataset. However, better parameter estimation could substantially increase system accuracy. Goldwater and Griffiths (2007) describe a method for integrating hyperparameter estimation into the Gibbs sampling procedure using a prior over possible values. Such an approach could be adopted in our framework, as well, and extended to include the layer weighting parameters. In addition, the infinite LDA model (Teh et al., 2006) automatically determines the optimal number of sense-clusters as an intrinsic part of the inference process. Adding both these components to our model would provide an elegant solution to the parameter estimation problem, and eliminate the need for tuning datasets and other external methods, such as cluster-validation. Finally, our Layered-LDA model represents a general extension of the LDA model, designed to be used wherever multiple sources of information are available. It would be interesting to apply this model to other tasks which conform to this setting. Possible examples include classification of scientific documents (where images and the abstract are possible additional layers to the main text), induction of music categories (where lyrics and different musical information can be viewed as layered elements of a song), and many others.

Bibliography

Agirre, Eneko and Oier Lopez de Lacalle. 2003. Clustering wordnet word senses. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*. pages 121–130.

Agirre, Eneko, Lluís Màrquez, and Richard Wicentowski, editors. 2007. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. The Association for Computational Linguistics, Prague, Czech Republic.

Agirre, Eneko and David Martínez. 2004. The basque country university system: English and basque tasks. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain, pages 44–48.

Agirre, Eneko, David Martínez, Oier López de Lacalle, and Aitor Soroa. 2006. Evaluating and optimizing the parameters of an unsupervised graph-based wsd algorithm. In *Proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing*. New York City, pages 89–96.

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)*. Prague, Czech Republic, pages 7–12.

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

Bar-Hillel, Yehoshua. 1960. The present status of automatic translation of languages. In Franz L. Alt, editor, *Advances in Computers I*, Academic Press, pages 91–163.

Barnard, K., P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. 2003. Matching words and pictures. *Journal of Machine Learning Research* 3(6):1107–1135.

Barzilay, R. and M. Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization (ISTS'97)*. Madrid, Spain, pages 10–17.

Berry, Michael W., Susan T. Dumais, and Gavin W. O'Brien. 1994. Using linear algebra for intelligent information retrieval. *SIAM Review* 37(4):573–595.

Bhattacharya, Indrajit and Lise Getoor. 2006. A latent dirichlet model for unsupervised entity resolution. In *The Society for Industrial and Applied Mathematics International Conference on Data Mining (SIAM-SDM)*. Bethesda, MD, USA, pages 47–58.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Bordag, Stefan. 2006. Word sense induction: Triplet-based clustering and automatic evaluation. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (ACL)*. pages 137–144.

Boyd-Graber, Jordan and David Blei. 2007. Putop: Turning predominant senses into a topic model for word sense disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pages 277–281.

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

Briscoe, Ted and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*. Las Palmas, Gran Canaria, pages 1499–1504.

Brody, Samuel and Mirella Lapata. 2008. Good neighbors make good senses: Exploiting distributional similarity for unsupervised WSD. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling)*. Manchester, UK, pages 65–72.

Brody, Samuel, Roberto Navigli, and Mirella Lapata. 2006. Ensemble methods for unsupervised wsd. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL/Coling)*. Sydney, Australia, pages 97–104.

Budanitsky, A. and G. Hirst. 2004. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics* 1(1):1–49.

Budanitsky, Alexander and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL 2001 Workshop: on WordNet and other lexical resources: Applications, extensions, and customizations*. Pittsburgh, PA, pages 29–34.

Cai, J. F., W. S. Lee, and Y. W. Teh. 2007. Improving word sense disambiguation using topic features. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-coNLL)*. pages 1015–1023.

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

Clear, Jeremy H. 1993. The british national corpus. In *The digital word: text-based computing in the humanities*, MIT Press, Cambridge, MA, USA, pages 163–187.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. 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)*. Morristown, NJ, USA, pages 56–63.

Daumé III, Hal. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name\#daume04cg-bfgs`.

Dietterich, T. G. 1997. Machine learning research: Four current directions. *AI Magazine* 18(4):97–136.

Edmonds, Philip. 2000. Designing a task for SENSEVAL-2. Technical note.

Edmonds, Philip and Adam Kilgarriff. 2002. Introduction to the special issue on evaluating word sense disambiguation systems. *Natural Language Engineering* 8(4):279–291.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Florian, Radu, Silviu Cucerzan, Charles Schafer, and David Yarowsky. 2002. Combining classifiers for word sense disambiguation. *Natural Language Engineering* 1(1):1–14.

Gale, W., K. Church, and D. Yarowsky. 1992a. A method for disambiguating word senses in a large corpus. In *Computers and the Humanities*. 26, pages 415–439.

Gale, William A., Kenneth W. Church, and David Yarowsky. 1992b. One sense per discourse. In *Proceedings of the HLT workshop on Speech and Natural Language*. Morristown, NJ, USA, pages 233–237.

Galley, Michel and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, pages 1486–1488.

Geman, S. and D. Geman. 1984. Stochastic relaxation, gibbs distribution, and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6):721–741.

Goldwater, Sharon. 2007. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

Goldwater, Sharon and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*. Prague, Czech Republic, pages 744–751.

Graff, David. 1995. North american news text corpus. Linguistic Data Consortium. LDC95T21.

Griffiths, Thomas L., Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, MIT Press, Cambridge, MA, pages 537–544.

Griffiths, Tom L. and Mark Steyvers. 2002. A probabilistic approach to semantic representation. In *Proeedings of the 24th Annual Conference of the Cognitive Science Society*. Fairfax, VA, pages 381–386.

Harris, Z. 1985. Distributional structure. *Katz, J. J. (ed.) The Philosophy of Linguistics* pages 26–47.

Hearst, Marti A. and Hinrich Schütze. 1993. Customizing a lexicon to better suit a computational task. In *Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*. pages 55–69.

Hoste, Vèronique, Anne Kool, and Walter Daelemans. 2001. Classifier optimization and combination in the english all words task. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (Senseval-2)*. Toulouse, France, pages 83–86.

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 North American Chapter of the Association for Computational Linguistics (HLT/NAACL), Companion Volume: Short Papers*. New York City, USA, pages 57–60.

Hsu, C. and C. Lin. 2001. A comparison of methods for multi-class support vector machines.

Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of 10th International Conference on Research in Computational Linguistics*. Taiwan.

Karov, Y. and S. Edelman. 1996. Learning similarity based word sense disambiguation from sparse data.

Kilgarriff, Adam. 2001. English lexical sample task description. In *Proceedings of SENSEVAL-2*. pages 17–20.

Kilgarriff, Adam and Martha Palmer. 2000. Introduction to the special issue on senseval. *Computers and the Humanities* 34:1–13.

Koeling, Rob, Diana McCarthy, and John Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of Human Language*

Technology Conference and Conference on Empirical Methods in Natural Language Processing. Vancouver, Canada, pages 419–426.

Kucera, Henry and W. Nelson Francis. 1967. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI, USA.

Landes, Shari, Claudia Leacock, and Randee I Tengi. 1998. Building semantic concordances. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, pages 199–216.

Langley, Pat, Wayne Iba, and Kevin Thompson. 1992. An analysis of bayesian classifiers. In *AAAI*. pages 223–228.

Leacock, Claudia, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics* 24(1):147–165.

Lee, Lilian. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL)*. College Park, MD, pages 25–32.

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 ACL-02 conference on Empirical methods in natural language processing*. Morristown, NJ, USA, 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 SIGDOC*. New York, NY, USA, pages 24–26.

Lin, Dekang. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics (Coling)*. Morristown, NJ, pages 768–774.

Lin, Dekang. 1998b. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference On Machine Learning (ICML)*. Madison, WI, pages 296–304.

Marcus, Mitchell P., Mary A. Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Journal of Computational Linguistics* 19(2):313–330.

McCarthy, Diana, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42th Annual Meeting of the Association of Computational Linguistics (ACL)*. Barcelona, Spain, pages 280–287.

Mihalcea, Rada. 2005a. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Conference of Human Language Technology and the European Chapter of the Association for Computational Linguistics (EMNLP)*. Vancouver, pages 411–418.

Mihalcea, Rada. 2005b. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Conference of Human Language Technology and the European Chapter of the Association for Computational Linguistics (EMNLP)*. Vancouver, pages 411–418.

Mihalcea, Rada and Phil Edmonds, editors. 2004. *Proceedings of the SENSEVAL-3*. Barcelona.

Mihalcea, Rada F. 2002. Word sense disambiguation with pattern learning and automatic feature selection. *Nat. Lang. Eng.* 8(4):343–358.

Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography* 3(4):235–244.

Miller, George A., Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the ARPA Human Language Technology Workshop*. Morgan Kaufman, pages 303–308.

Mohammad, Saif and Graeme Hirst. 2006. Determining word sense dominance using a thesaurus. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EMNLP)*. Trento, pages 113–120.

Moldovan, Dan I. and Rada Mihalcea. 2000. Using wordnet and lexical operators to improve internet searches. *IEEE Internet Computing* 4(1):34–43.

Mooney, Raymond J. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Somerset, New Jersey, pages 82–91.

Morris, Jane and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 1(17):21–43.

Murray, Gabriel, Steve Renals, Jean Carletta, and Johanna Moore. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of the Conference of Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*. New York City, USA, pages 367–374.

Navigli, Roberto. 2005. Semi-automatic extension of large-scale linguistic knowledge bases. In *Proceedings of the 18th FLAIRS*. Florida, pages 548–553.

Navigli, Roberto. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL/Coling*. Sydney, Australia, pages 105–112.

Navigli, Roberto and Paola Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *Journal of the IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence* 27(7):1075–1088.

Ng, Hwee Tou and Hian Beng Lee. 1997. Dso corpus of sense-tagged english. Linguistic Data Consortium. LDC97T12.

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 on Association for Computational Linguistics (ACL)*. Morristown, NJ, USA, pages 455–462.

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

Nigam, K., J. Lafferty, and A. McCallum. 1999. Using maximum entropy for text classification.

Niu, Cheng, Wei Li, Rohini K. Srihari, and Huifeng Li. 2005a. Word independent context pair classification model for word sense disambiguation. In *Proceedings*

of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005). Ann Arbor, Michigan, pages 33–39.

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

Niu, Zheng-Yu, Dong-Hong Ji, and Chew-Lim Tan. 2007. I2r: Three systems for word sense discrimination, chinese word sense disambiguation, and english word sense disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pages 177–182.

Palmer, Martha, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1):71–106.

Pantel, Patrick and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM Special Interest Group on Knowledge Discovery in Data (SIG-KDD)*. ACM Special Interest Group on Knowledge Discovery in Data, New York, NY, USA, pages 613–619.

Pedersen, T., S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) Demonstrations*. Boston, MA, pages 38–41.

Peirce, Charles Sanders. 1933. The simplest mathematics. In Charles Hartshorne and Paul Weiss, editors, *Collected Papers of Charles Sanders Peirce*, Harvard University Press, Cambridge, MA, volume 4.

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)*. Prague, Czech Republic, pages 87–92.

Preiss, Judita. 2004. Probabilistic word sense disambiguation. *Computer Speech & Language* 18(3):319–337.

Preiss, Judita and David Yarowsky, editors. 2001. *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*. Toulouse, France.

Purandare, Amruta and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of CoNLL-2004*. pages 41–48.

Ramakrishnan, Ganesh, Apurva Jadhav, Ashutosh Joshi, Soumen Chakrabarti, and Pushpak Bhattacharyya. 2003. Question answering via bayesian inference on lexical relations. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*. Morristown, NJ, USA, pages 1–10.

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Somerset, New Jersey, pages 133–142.

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

Slonim, Noam, Nir Friedman, and Naftali Tishby. 2002. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA, pages 129–136.

Snow, Rion, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. 2007. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. pages 1005–1014.

Stokoe, Christopher. 2005. Differentiating homonymy and polysemy in information retrieval. In *Proceedings of the Conference of Human Language Technology and the European Chapter of the Association for Computational Linguistics (HLT/EMNLP)*. pages 403–410.

Strapparava, Carlo, Alfio Gliozzo, and Claudio Giuliano. 2004. Word-sense disambiguation for machine translation. In *Proceedings of the SENSEVAL-3*. Barcelona, pages 229–234.

Tan, Pang-Ning and Rong Jin. 2004. Ordering patterns by combining opinions from multiple sources. In *Proceedings of the 10th Conference on Knowledge Discovery and Data Mining*. Seattle, WA, pages 22–25.

Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101(476):1566–1581.

Toutanova, Kristina and Mark Johnson. 2008. A bayesian lda-based model for semi-supervised part-of-speech tagging. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, MIT Press, Cambridge, MA, pages 1521–1528.

van Halteren, Hans, Jakub Zavrel, and Walter Daelemans. 2001. Improving accuracy in word class tagging through combination of machine learning systems. *Computational Linguistics* 27(2):199–230.

Véronis, Jean. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language* 18(3):223–252.

Vickrey, David, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proceedings of the Conference of Human Language Technology and the European Chapter of the Association for Computational Linguistics (HLT/EMNLP)*. Vancouver, pages 771–778.

Voorhees, Ellen M. 1993. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, pages 171–180.

Vossen, Piek, editor. 1998. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA.

Weaver, Warren. 1949/1955. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, MIT Press, Cambridge, MA, pages 15–23. Reprinted from a memorandum written by Weaver in 1949.

Weeds, Julie. 2003. *Measures and Applications of Lexical Distributional Similarity*. Ph.D. thesis, University of Sussex.

Widdows, Dominic. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In Marti Hearst and Mari Ostendorf, editors, *Proceedings of the Conference of Human Language Technology and the*. Edmonton, Alberta, Canada, pages 276–283.

Yarowsky, D. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proc. of the 14th International Conference on Computational Linguistics (Coling)*. Nantes, France, pages 454–460.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics (ACL)*. pages 189–196.

Yarowsky, David. 2000. Word sense disambiguation. In R. Dale, H. Moisl, and H. Somers, editors, *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, Marcel Dekker Inc., New York, pages 629–654.

Yarowsky, David and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering* 9(4):293–310.

Zhu, X. and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02.