Title
Making the Most of It: Word Sense Annotation and Disambiguation in the Face of Data Sparsity and Ambiguity

Author
Jurgens, David Alan

Publication Date
2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Making the Most of It:

Word Sense Annotation and Disambiguation in the Face of Data Sparsity and Ambiguity

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

David Alan Jurgens

2014

ABSTRACT OF THE DISSERTATION

Making the Most of It:

Word Sense Annotation and Disambiguation in the Face of Data Sparsity and Ambiguity

by

David Alan Jurgens

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2014

Professor Michael Dyer, Chair

Natural language is highly ambiguous, with the same word having different meanings depending on the context. While human readers often have no trouble interpreting the correct meaning, semantic ambiguity poses a significant problem for many natural language systems, such as those that translate text or perform machine reading. The task of identifying which meaning of a word is present in a given context is known as Word Sense Disambiguation (WSD), where a word's meanings are discretized into units referred to as *senses*. Because languages contain hundreds of thousand of unique words and each of those words can have multiple meanings, comprehensive sense-annotated corpora are often sparse, with only tens to low-hundreds of annotated examples of each word. As a result, creating high performance WSD systems requiring overcoming this data sparsity.

This thesis provides a three-fold approach to improving WSD performance in the face of data sparsity. First, we introduce two new algorithms that take the role of a lexicographer and automatically learn the senses of a word from example uses in a fully unsupervised way. We then demonstrate that these unsupervised systems can be combined with a limited amount of annotated data to create a semi-supervised WSD

system that significantly outperforms a state-of-the-art supervised WSD system trained on the same data. Second, we propose a novel method for gathering high-quality sense annotations from large numbers of untrained, online workers, commonly referred to as crowdsourcing. Our method lowers the time and cost of building sense-annotated corpora, while maintaining as high a level of agreement between annotators, comparable with that of trained experts. Third, we analyze cases of ambiguity in sense annotations, when two annotators differ about which sense best describes the meaning of a particular usage of a word. To perform this analysis, we built the largest sense-annotated corpus where cases of semantic ambiguity are explicitly marked. Our analysis of this corpus revealed multiple causes for this ambiguity as well as how the ambiguity may be interpreted and resolved by natural language applications using ambiguous data. To complement this work on ambiguity, we have also introduced a new methodology for evaluating WSD systems that explicitly report ambiguous instances.

The dissertation of David Alan Jurgens is approved.

Junghoo Cho

D. Stott Parker, Jr.

Jessica Rett

Michael Dyer, Committee Chair

University of California, Los Angeles

2014

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGMENTS

First, I thank my family an friends for their support during the endeavor. Grad schools is a trip that no one takes alone, and I could not have done this without you. You made the journey worth it. I am also eternally grateful for the support of the Waldemar Estate.

Many thanks to my advisor, Michael Dyer who consistently challenged me to be a better researcher and from whom I learned so much.

Finally, I have the good fortune to work with and learn from some amazing collaborators and I owe them my deepest gratitude. In rough order of appearance, I especially thank the following: Seth Proctor, James Megquier, Jim Waldo, Keith Stevens, Uri Schonfeld, Michael Shindler, Keith Holyoak, Saif Mohammad, Peter Turney, Tsai-Ching Lu, David Allan, Ioannis Klapaftis, Taher Pilehvar, Roberto Navigli, and Osman Baskaya. The great discussions and collaborations with each of you made this work possible.

Chapter 3 and parts of Chapter 6 are based on work with Osman Baskaya which is under preparation. Chapter 6 is based in part on Jurgens and Klapaftis (2013). Chapter 5 was supported in part by ERC Starting Grant Multi-JEDI No. 259234 for which Roberto Navigli is the Principle Investigator.

VITA

2004	B.A. Philosophy, Washington University in St. Louis.
2004	M.S. Computer Science, Washington University in St. Louis.
2004–2005	Software Development Engineer, Amazon.com, Seattle, WA.
2005–present	Doctoral Student, UCLA, Los Angeles, CA.
2006–2010	Teaching Assistant, UCLA, Los Angeles, CA.
2006, Summer	Research Intern, Sun Microsystems Laboratories, Burlington, MA.
2007, Summer	Research Intern, Sun Microsystems Laboratories, Burlington, MA.
2008, Summer	Research Intern, Sun Microsystems Laboratories, Burlington, MA.
2010–2012	Visiting Researcher, HRL Laboratories, Malibu, CA.
2013–2014	Research Scientist, Sapienza University of Rome, Rome, Italy.

PUBLICATIONS

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli, "SemEval-2014 Task 3: Cross-level Semantic Similarity." Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval), 2014. ACL.

Daniele Vannella, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli, "Validating and Extending Semantic Knowledge Bases using Video Games with a Purpose." Proceedings of the Annual Meeting for the Association for Computational Linguistics (ACL), 2014. ACL.

David Jurgens. "An analysis of ambiguity in word sense annotations." Proceedings of the International Conference on Language Resources and Evaluation (LREC), 2014.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli, "Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity." Proceedings of the Annual Meeting for the Association for Computational Linguistics (ACL), 2013. ACL.

David Jurgens, "That's what friends are for: Inferring location in online communities based on social relationships." Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM), 2013. AAAI.

Roberto Navigli, David Jurgens, and Daniele Vanilla, "SemEval-2013 Task 12: Multilingual Word Sense Disambiguation." Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval), 2013. ACL.

David Jurgens and Ioannis Klapaftis, "SemEval-2013 Task 13: Graded Word Sense Induction" Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval), 2013. ACL.

David Jurgens. "Embracing Ambiguity: A Comparison of Annotation Methodologies for Crowdsourcing Word Sense Labels." Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL), 2013. ACL.

Veronika Strnadova, David Jurgens, Tsai-Ching Lu. "Characterizing Online Discussions in Microblogs Using Network Analysis." Proceedings of the AAAI Spring Symposium on Analyzing Microtext, 2013. AAAI.

David Jurgens, Saif M. Mohammad, Peter D. Turney, and Keith J. Holyoak. "SemEval-2012 Task 2: Measuring Degrees of Relational Similarity." Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012). ACL.

David Jurgens, "An Evaluation of Graded Sense Disambiguation using Word Sense Induction." Proceedings of *SEM, the First Joint Conference on Lexical and Computational Semantics, 2012. ACL.

David Jurgens and Tsai-Ching Lu, "Friends, Enemies, and Lovers: Detecting Communities in Networks Where Relationships Matter" Proceedings of Web Science, 2012. ACM.

David Jurgens and Tsai-Ching Lu, "Temporal Motifs Reveal the Dynamics of Editor Interactions in Wikipedia." Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM), 2012. AAAI.

David Jurgens and Keith Stevens, "Measuring the Impact of Sense Similarity on Word Sense Induction." Proceedings of the EMNLP 2011 Workshop on Unsupervised Learning in NLP, 2011. ACL.

David Jurgens, "Word Sense Induction by Community Detection." Proceedings of the ACL 2011 Workshop TextGraphs-6: Graph-based Methods for Natural Language Processing, 2011. ACL.

David Jurgens and Keith Stevens. "Capturing Nonlinear Structure in Word Spaces Through Dimensionality Reduction." Proceedings of the ACL 2010 Workshop on GEometrical Models of Natural Language Semantics (GEMS), 2010.

David Jurgens and Keith Stevens. "HERMIT: Flexible Clustering for the SemEval-2 WSI Task." Proceedings of the ACL 2010 SemEval-2010 Workshop, 2010.

David Jurgens and Keith Stevens. "The S-Space Package: An Open-Source Framework for Word Space Algorithms." Proceedings of the ACL 2010 System Demonstrations, 2010.

David Jurgens and Keith Stevens. "Event Detection in Blogs using Temporal Random Indexing." In *Proceedings of the International Workshop on Events on Emerging Text Types (eETTs)*, pages 21-28, 2009.

Robert Pless and David Jurgens. "Road extraction from motion cues in aerial video." In *Proceedings of the ACM Conference on Geographic Information Systems*, pages 31-38, 2004.

CHAPTER 1

Introduction

Understanding the meaning of natural language often requires understanding the meaning of the words being used. For example, consider the meaning of the term *bass* in the following sentences:

- I went fishing and caught a bass.
- I sing bass in the choir.

In the first sentence, *bass* refers to a fish, while in the second, it refers to a type of singing voice. Representing and understanding the meaning of both sentences is dependent on correctly interpreting *bass*. This interpretation is equally essential for tasks such as translation and summarization that must transform the text itself based on its meaning.

The task of identifying the meaning of a word in a given context is known as Word Sense Disambiguation (WSD).[1] Here, a sense refers to a distinct meaning of a word. Research on WSD began as a part of work on machine translation (Weaver, 1949) and has since been incorporated by many other areas of Natural Language Processing (NLP) to improve their performance. For example, our ADW algorithm measures the similarity of two sentences by first disambiguating the text and representing its meaning using senses (Pilehvar et al., 2013). Comparing our sense-based representation of the sentences' meanings to measure their similarity achieves state-of-the-art performance

[1]For clarity, we always use the term *disambiguate* for cases when a WSD system (i.e., a software program) is determining which sense is present and use the term *annotate* when a human is performing the analogous task.

in matching human similarity judgments, surpassing the performances of current approaches, most of which do not disambiguate the text.

Word sense disambiguation is considered a core task of NLP (Ide and Véronis, 1998; Navigli, 2009) and WSD systems have provided a significant benefit to many NLP areas such as information retrieval (Navigli and Crisafulli, 2010; Zhong and Ng, 2012), paraphrasing (Rus et al., 2009), sentiment analysis (Rentoumi et al., 2009; Martın-Wanton et al., 2010; Balamurali et al., 2011), knowledge extraction (Hassan et al., 2006; Ciaramita and Altun, 2006; Navigli and Ponzetto, 2010; Hartmann et al., 2013), and semantic role labeling (Che et al., 2010). Furthermore, despite performance issues with sense disambiguation in the early years of machine translation, WSD has since proven beneficial to more recent machine translation systems (Dagan and Itai, 1994; Vickrey et al., 2005; Carpuat and Wu, 2007; Chan et al., 2007). Given the wide-spread use of WSD systems, further improvements to WSD performance can offer corresponding benefits to many areas of NLP.

Two central challenges face WSD systems. The first challenge is the sparsity of sense annotated data. Typically, word uses are annotated with senses from an established sense inventory, such as WordNet (Fellbaum, 1998) or OntoNotes (Hovy et al., 2006). These inventories provide clear descriptions of each sense's meaning and occasionally syntactic information on how the sense is used. However, because of the large number of unique words in a language and resources required to produce annotated examples of each, words are usually associated with only tens of sense-annotated examples, with few words having more than several hundred annotated instances. For example, the largest sense-annotated resource, SemCor (Miller et al., 1993), contains only 234,157 annotated examples. Because of this sparsity, many WSD approaches have incorporated techniques such as unsupervised methods that use knowledge from the sense inventory itself to disambiguate (e.g., (Pedersen and Kolhatkar, 2009) and (Navigli and Lapata, 2010)) or methods for automatically generating more examples of sense-associated features (e.g., (Agirre et al., 2001) and (Zhong and Ng, 2010)). How-

ever, even with these additions, supervised methods often only sightly out-perform the baseline of always selecting the most-frequent sense of a word (Navigli et al., 2007; Pradhan et al., 2007), with unsupervised approaches generally performing worse than supervised approaches.

The second challenge stems from the difficulty of the task itself and distinguishing between senses. In the easiest setting, a word is a homonym and has unrelated meanings, such as *bass* with its fish and vocal senses and *bank* with its senses of a financial institution and a sloping surface that borders water. Because these senses are unrelated, they are generally easier to disambiguate using the specific contextual cues associated with each (e.g., "fishing" or "music" for the senses of *bass*); furthermore, these cues are easily learned by WSD systems because of distinctness of the contexts in which the unrelated senses appear. However, many words have senses that are related in some way. Extending our earlier example, *bass* may be both a singing voice and a musical instrument (e.g., a double bass). This sense relatedness results in two challenges. First, because of their relatedness, both meanings may appear in similar music-related contexts, which increases the difficulty of learning features that distinguish between them. This difficulty is further compounded by the sparsity of sense-annotated examples, which could potentially reveal key contextual differences between such senses. Second, in some contexts, the senses may be indistinguishable from one another. For example, in the context "The *bass* was too loud at the concert last night." *bass* clearly refers to a musical entity and not the fish, but the context itself is ambiguous as to which musical entity. This second challenge also causes issues for human annotators when creating sense-annotated corpora and results in annotator disagreements, which increase the time and costs required for producing such corpora (Palmer et al., 2007).

In this thesis, we address both challenges in three ways. First, we show how the sparse sense-annotated data can be more effectively used by combining the data with the results of unsupervised algorithms that automatically learn a word's meanings in order to produce a semi-supervised WSD system. We demonstrate that this semi-supervised

system offers superior performance to fully supervised WSD state of the art in settings where data is sparse. Second, we propose new methods for reducing data sparsity by producing sense-annotated data from untrained workers on crowdsourcing platforms, showing that our methods result in expert-level agreement on the resulting annotations. Third, we analyze the challenge of ambiguity when sense annotating in order to identify the causes behind multiple interpretations, their frequencies, and how the annotation process and WSD system evaluations can be improved in light of this ambiguity.

Following, in Section 1.1, we describe our contributions towards these problems in more detail and then, in Section 1.2, we present the outline of this thesis. Last, in Section 1.3, we formalize the terminology used in this thesis for the benefit of the reader.

1.1 Contributions

This thesis provides contributions in four categories.

1.1.1 Semi-supervised WSD

Given the limited amount of training data, we propose a new semi-supervised approach to performing WSD, which makes better use of the annotated data than using it directly to train a supervised system. We describe a new method for building semi-supervised WSD systems that combines sense-annotated data with the output of fully-unsupervised methods that learn the senses of words automatically. These unsupervised methods, known as Word Sense Induction (WSI) methods, examine how a word is used throughout a corpus in order to learn its different meanings and how to distinguish between them. The information learned by the WSI model about a word's different meanings is combined with annotated examples to create a semi-supervised system that is capable of disambiguating new instances of the word in contexts that contain features never seen in the original sense-annotated examples.

Our work on sense induction offers three contributions. First, we propose two new methods for inducing senses from graphs of semantic relations and demonstrate that they identify clear semantic distinctions. Second, we evaluate the current method of constructing semi-supervised WSD systems from sense induction methods and show that our novel approach produces a statistically significant improvement over both the current method and over state-of-the-art supervised WSD. Third, we examine the effect of the amount of data for training either semi-supervised or fully-supervised WSD systems and identify the point at which one out-performs the other. This third contribution enables researchers to properly optimize their WSD systems based on the amount of annotated data.

1.1.2 Sense Annotation by Non-experts

The problem of data sparsity may also be addressed by finding alternate, more efficient or cost-effective methods to sense annotate data. The current methods of using experts do not scale due to the time-constraints and limited supply of experts; similarly, methods using trained annotators are often slow due to the time and costs involved in training.

Our work examines using untrained workers to produce sense annotations. Specifically, we focused on recruiting workers from crowdsourcing platforms such as Amazon Mechanical Turk and CrowdFlower, which enables gathering annotations from potentially tens of thousands of individuals and provides the much-needed scale. We propose two new annotation methodologies that are specifically adapted to the crowdsourcing setting to match the workers' unfamilarity with the task and their ambiguity in assigning senses. Our contribution is showing that one of our new methodologies enables workers to produce high-quality annotations with annotation agreement rates on par with those seen in expert-annotated corpora. In contrast, traditional annotation methodologies used by experts result in poor worker performance. As a result, we demonstrate that crowdsourcing offers a potential way to mitigate data sparsity by gathering high-

quality annotations cheaply.

1.1.3 Understanding Causes of Annotation Difficulty

Sense annotation is often difficult due to the relatedness of a word's senses, as was illustrated for *bass*. Typically, this difficulty is evidenced by disagreements between human annotators during the annotation process over which sense applies. Significant effort has been put into revising a word's sense definitions to avoid confusion and, ideally, reduce the number of annotator disagreements (Edmonds and Cotton, 2001; Palmer et al., 2004; Hovy et al., 2006). However, little work has examined the underlying causes for *why* annotators disagree. In some instances, a word's use is genuinely ambiguous, either due to too little context, or from having contextual features that evoke multiple distinct interpretations of the use. In such cases, annotators may have valid -though differing- interpretations, which should not be treated as a case of disagreement.

We provide two contributions towards understanding some of the most difficult cases seen in sense annotation and disambiguation. First, we analyze the uses of twenty nouns, twenty verbs, and ten adjectives in a total of 4664 contexts to identify the frequency with which each might have more than one valid sense interpretation. This analysis produced the largest publicly-available sense-annotated corpus in which instances may be annotated with multiple senses, showing that, on average, 11% of instances can have multiple valid interpretations. Second, we analyze the instances with multiple interpretations to identify the underlying cause of the interpretations, showing that contextual underspecification is the predominant cause of ambiguity. This finding suggests that future work on WSD could improve its performance by recognizing underspecified contexts and then seeking additional context outside of the immediate sentence containing the usage. As a part our analysis, we propose a new classification for specifying the type of ambiguity that is present and how the word may be interpreted based on that ambiguity. From this classification, we provide two recommendations for improving the guidelines used by future sense annotation efforts.

1.1.4 Evaluation Measure for Comparing Ambiguous Annotations

In some cases of ambiguity, the meaning of a word use is best modeled with multiple senses. Current evaluations for WSD systems have used only instances with a single sense label and their metrics have been designed for such single-sense instances, making them unsuitable for use in the multiple sense setting. Therefore, we evaluate extensions of current metrics and new metrics for both WSD and WSI systems in the multiple-sense setting. We demonstrate cases where certain metrics are unable to identify problematic annotations and where some metrics have systematic biases. Our contribution is identifying complementary, robust metrics for evaluating WSI and WSD systems in this new setting.

1.2 Thesis Overview

We begin by introducing Word Sense Induction in Chapter 2 and present two algorithms for automatically learning the meaning of words, based on our work in Jurgens (2011) and Jurgens (2012). We demonstrate examples of features associated with induced senses and contexts tagged with each sense in order to highlight the induction and disambiguation capabilities of each system.

In Chapter 3, we show how WSI systems can be used to build high-performance semi-supervised WSD systems. Instead of using the context around a usage as features for disambiguation, these semi-supervised systems first label the usage with induced senses and then used the induced senses as features for disambiguating the usage with senses from an established sense inventory such as WordNet (Fellbaum, 1998). We present a new method for building semi-supervised systems that out-performs current state of the art. Using our new method, we show how multiple WSI models may be combined into a single ensemble semi-supervised WSD system, which produces a large statistically-significant improvement over any one model. To test and demonstrate the systems' effectiveness, we introduce a new evaluation setting for semi-supervised WSD

systems that are tested on two orders of magnitude more data in precisely-controlled conditions, enabling a more reliable estimation of performance than used in current tests. Finally, we show that the ensemble semi-supervised system provides superior performance to fully supervised WSD systems in cases where fewer than several hundred annotated instances are available, which is the case for most words for which annotated data is available.

In Chapter 4, we consider an alternate approach for addressing data sparsity by showing how untrained, crowdsourced workers can be used to gather annotations at scale instead of relying solely on trained lexicographers. Starting from our work in Jurgens (2013), we introduce issues in sense annotation and highlight the challenges in making fine-grained sense distinctions. We describe our two new methodologies for gathering sense annotations and compare them with current annotation methods. We then demonstrate that our methods enable workers to equal the inter-annotator agreement rates of experts in current annotated corpora, which was not seen with current methods.

Based on the challenges of word sense annotation, in Chapter 5, we analyze annotations with respect to ambiguity (i.e., whether two senses seem equally-good annotations for the same usage), which is a core problem faced by annotators and WSD systems. We begin by describing our effort in producing a new sense-annotated corpus in Jurgens and Klapaftis (2013), which contains instance that have been explicitly marked with all applicable WordNet senses. This corpus is the largest corpus of its kind. Following, we present our analysis, described in Jurgens (2014), where we analyze each instance having multiple sense annotations to identify the causes of such ambiguity and those causes' frequencies. Our analysis reveals three major findings. First, we show that too little informative context is responsible for the majority of ambiguous cases. Second, we show that the fine-grained sense distinctions of WordNet are not completely responsible for the perception of multiple senses. Last, we propose a two-way classification scheme, which we argue would serve sense annotators as a guideline when considering

multiple senses for a usage.

As a formal quantitative comparison, in Chapter 6, we present a series of experiments that demonstrate the state-of-the-art performance of our WSI and semi-supervised WSD models. This chapter contains the results of SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013), which we co-organized as a way of presenting a standardized evaluation methodology for WSI and WSD models on instances that are labeled with multiple word senses. We include an additional performance comparison analysis of existing unsupervised and unsupervised WSD systems on the task, showing that our semi-supervised models offer superior performance. In a second set of evaluations, we test our ensemble semi-supervised WSD system design on our SemEval-2013 Task 13 and prior SemEval WSI tasks, showing that it provides a consistent performance improvement.

Last, in Chapter 7, we synthesize and review the research contributions of this thesis. Following, we discuss our on-going work to extend these contributions. To conclude, we highlight several open questions that remain for research on Word Sense Disambiguation and Induction.

1.3 Terminology

Briefly, we clarify some of the terminology and notation used throughout this thesis. A token refers to an occurrence of a word in a context. We use the notation "token" when describing a use of a token in text. A word type denotes a concept that can be expressed in different morphological forms (i.e., tokens). For example, the type *run* may be expressed in tokens such as "run," "running," and "ran." Additionally, some types such as Named Entities or fixed expressions (e.g., idioms) may span several tokens, e.g., "White House," "pain in the neck," or "lead on" (i.e., deceive), where the tokens as a whole denote the type.

For stylistic reasons, we may refer to a type as term or lemma with no difference

in meaning; where unclear, we will always use type. Types are denoted using italics, e.g., *bass*. When the part of speech is ambiguous, we will append a type with its part of speech, e.g., *lead.n* or *address.v*.

A usage denotes the occurrence of a particular word type in a context; for stylistic reasons we will occasionally refer to a usage as use or instance when the context is clear. When annotated, a usage will have one or more sense labels, which each label denotes a specific sense. We refer to a set of labels for multiple usages as a sense labeling. For example, if two WSD systems label the same corpus, we say that the corpus has two sense labelings. If a usage is labeled with more than one sense, we use the description multi-sense.

CHAPTER 2

Learning and Disambiguating Word Senses from Word Communities

Summary Performance on Word Sense Disambiguation is often constrained by the availability of sense-annotated data. Rather than rely solely on human-annotated data, an alternate approach has been to use fully-unsupervised systems that learn a word's senses directly from how the word is used and that are able to produce automatically-disambiguated data. This chapter introduces two new unsupervised techniques for automatically learning word senses based on our work in Jurgens (2011) and Jurgens (2012), which learn senses from networks of semantic relations. We illustrate the learning process of both our methods and then show examples of the induced senses, demonstrating that the methods are effective at automatically identifying meaningful sense distinctions.

2.1 Introduction

Work on Word Sense Disambiguation depends on the availability of sense-annotated data, both for building supervised systems and for evaluating the quality of all types of WSD systems. However, sense annotation is an intensive process, requiring both significant time and human resources (Navigli, 2009). The intensive nature of sense annotation creates a knowledge acquisition bottleneck in which WSD methods must learn from sparse data due to the high cost and time required to gather additional annotations (Gale et al., 1992). For example, in the largest sense-annotated corpus, SemCor

(Miller et al., 1993), all but 97 of the 11,685 polysemous lemmas in SemCor have fewer than 200 annotated instances, with many lemmas having just a few example instances of their less-common senses.[1] This sparsity of annotated examples presents a significant challenge to WSD methods, especially for less-common senses.

A potential solution to the knowledge acquisition bottleneck is to learn a word's senses directly from how it is used, which has the potential to disambiguate an arbitrary number of instances automatically using the learned senses. A word's uses are collected and then automatically organized such that uses having the same meaning are grouped together, a process often referred to as Word Sense Induction (WSI). This process is an analog of the one used by lexicographers when constructing resources such as dictionaries (Church and Hanks, 1990). However unlike the manual investigations of lexicographers, sense induction methods are completely unsupervised and use lexical and semantic regularities in a word's contexts to identify its different meanings. We illustrate this with an example; consider the meaning of *pescespada* in the following contexts:

- He grilled the pescespada well done.
- Pescespada goes well with capers and lemon.
- Catching a pescespada takes many hours in the open ocean.
- I used to drive a pescespada to work.
- She bought a used pescespada from the dealer.
- Fiat is releasing the 2014 model of the pescespada later this year.

The context of *pescespada* suggest two different senses: a type of fish and a type of car. Humans easily recognize these sense distinctions based on the contextual clues relating to *pescespada*'s meanings: the first three sentences contain words such as *grilled* and *ocean* known to be associated with fish, while the latter three sentences contain words such as *drive* and *Fiat* known to be associated with cars. By identifying these types of

[1] For example, *bass.n* occurs just three times, with all occurrences referring the music-related senses and no examples of the fish-related sense.

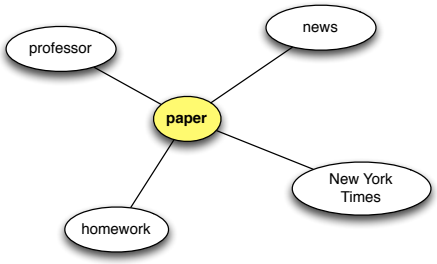
associations in each of a word's contexts, a sense induction method (1) learns regularities in the word's associations that each evoke each distinct meaning, and (2) labels each context with which meaning is present based on the associations, i.e., disambiguates the instance with one of the induced senses.

We propose two new methods for sense induction based on the one sense per collocation hypothesis (Yarowsky, 1993): Given a collocation consisting of two semantically associated word types (e.g., *bass* and *ocean*), the semantic association holds between only one sense of each word. Following convention, a collocation is defined as two words appearing together in text and sharing some semantic relationship.[2] In our method, we identify groups of collocations that frequently occur together and share relations with a word, which are used identify its different senses, e.g., discovering in the earlier *pescespada* example that the collocations of (*pescespada*, *ocean*), (*pescespada*, *catch*), and (*ocean*, *catch*) occur in similar contexts and point to the same meaning of *pescespada*.

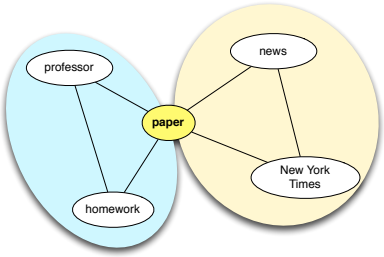
In our methods, sense induction is performed in two phases: (1) detecting words with strong semantic associations to a target word and forming collocations and (2) identifying the groups of related collocations that co-occur with the same sense of the target. Both of our methods accomplish the first phase by constructing a graph where edges connect statistically-associated lemmas. For the second phase, our first method clusters a graph to perform an all-words sense induction to identify groups of collocations that may be applied to disambiguate multiple words; in contrast, our second method induces senses from a graph for one word at a time.

We illustrate our methods' process abstractly for the target lemma *paper* using Figure 2.1. Figure 2.1a shows a graph for the target after four collocations are added. In their disconnected state, the collocations do not reveal any distinctions in the target's meaning. Therefore, edges are added between the collocations themselves when they are related with each other. Figure 2.1b shows two edges added to the graph, forming

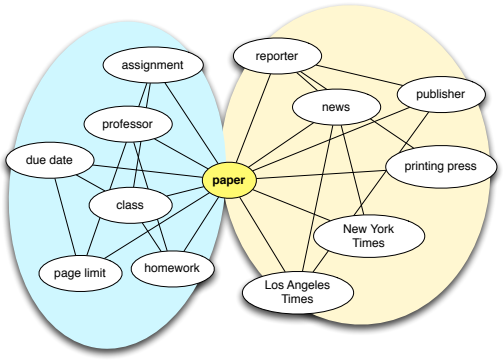
[2]Interpretations of this general definition are considered later in Section 2.2.1



(a) Collocations



(b) Relationships between collocations



(c) Clusters of related collocations

Figure 2.1: A high-level depiction of the sense induction process used in this thesis: (1) collocations are found for a target word and modeled as a graph, in Figure 2.1a, (2) then edges are added between the collocations themselves to discover densely-connected clusters that denote senses, shown in stages as Figures 2.1b and 2.1c.

two triads. The connected structure of these collocations suggests that the target has two meanings. A larger graph for the target with more collocations and edges, shown in Figure 2.1c, has a similar structure with two densely (but not completely) connected clusters of collocations. Our methods first build graphs that have this type of modular structure and then identify the groups of densely connected vertices in order to induce senses, shown as shaded ovals in the Figures 2.1b and 2.1c.

This chapter contains two main contributions. First, in Section 2.2, we review cur-

rent methods for identifying collocations, showing that many current methods are biased by word frequency and insufficient for the purposes of sense induction. We ultimately demonstrate that the χ^2 statistic offers the most robust way of identifying collocations for inducing senses. Second, in Section 2.3, we propose two new methods for sense induction from graphs of collocations and in Section 2.4 demonstrate that both methods identify meaningful sense distinctions for a variety of example words.

2.2 A Graph Representation of Collocations

Graph-based WSI operates by finding groups of related collocations using the collocations' connectivity in a graph structure. Graphs provide a natural way of modeling the relationships between collocations by representing word types as vertices and adding edges between types that participate in a collocation. The challenge is therefore to construct a graph whose structure has densely-connected regions that each correspond to a different meaning of a word. Furthermore, because words vary in the frequency with which they appear, the construction procedure should produce graphs with similar sizes or properties – especially if the method is parameterizable.

Constructing a collocation graph for WSI depends has two parameters: (1) how collocations are identified computationally and (2) which collocations are included in the graph itself. Following, we define four common methods for defining collocations in the graph (Sec. 2.2.1) and then describe three approaches to building graphs from these collocations (Sec. 2.2.2).

2.2.1 Modeling Collocations

Because of the loose definition of a collocation as two words occurring together in text with some defined relationship (Yarowsky, 1993), multiple approaches have defined the specifics of a collocation differently according to two parameters: (1) the conditions in which two words participate in a collocation and (2) the statistic for measuring the

strength of the collocation's relationship, where larger values indicate a closer semantic connection.[3]

The first choice of the scope of a collocation has varied widely between approaches. For example, Yarowsky (1993) suggests using two lemmas that either occur sequentially or are involved in direct syntactic relationship (e.g., subject-verb); in contrast, Véronis (2004) considers all pair-wise combinations of content words in a paragraph as collocations.[4] These two extremes represent a trade-off in the information that is represented in a collocation. More immediate scopes (i.e., the words to the left and right) are known to capture information on how the word is used, whereas larger scopes capture topical associations (Peirsman et al., 2008; Utsumi, 2010). Both sources of information are useful for sense induction, and so most approaches have adopted a middle ground of restricting the context to a single sentence and considering all pair-wise combinations of content words as collocations, e.g., (Dorow and Widdows, 2003; Bordag, 2006; Jurgens, 2011). We adopt the sentence boundary in this work.

The strength of a collocation has been frequently computed with one of four statistics: (1) co-occurrence frequency, (2) conditional probability, (3) the Dice coefficient and (4) the χ^2 statistic. We define these four methods next. Given two lemmas w_1 and w_2, let $c(w)$ and $c(w_1, w_2)$ denote the frequency of a lemma and the frequency of their co-occurrence together in a corpus. Raw co-occurrence frequency is simply the value $c(w_1, w_2)$ and has been used widely (Dorow and Widdows, 2003; Biemann, 2006; Jurgens, 2011). Véronis (2004) proposes using the maximum of the words' conditional probabilities: $max(p(w_1|w_2), p(w_2|w_1))$.[5] Navigli and Crisafulli (2010) and Manandhar et al. (2010) propose using the Dice coefficient, which is measured as $\frac{2c(w_1,w_2)}{c(w_1)+c(w_2)}$. In Jurgens (2012) we propose using the χ^2 statistical association, which is computed

[3]For a survey of collocation definition in general NLP settings, see Evert (2005).

[4]In the original version,Véronis (2004) includes only nouns and adjectives with minimal filtering of high-frequency lemmas from a stop list, but later applications of the algorithm have included verbs as well (Agirre et al., 2006a).

[5]In the original formulation, Véronis (2004) uses $1 - max(p(w_1|w_2), p(w_2|w_1))$; we invert the definition here without loss of generality to make the interpretation of the value consistent with that of the other statistics.

Co-occurrence	Conditional Probability	Dice	χ^2
be.v	largemouth.n	drum.n	drum.n
guitar.n	crappie.n	guitar.n	guitar.n
drum.n	smallmouth.n	vocal.n	vocal.n
band.n	tasmania's.n	keyboard.n	largemouth.n
play.v	bluegill.n	drum.v	keyboard.n
have.v	fretless.j	guitarist.n	drum.v
vocal.n	pumpkinseed.n	percussion.n	crappie.n
player.n	muskellunge.n	piano.n	percussion.n
also.r	panfish.n	band.n	smallmouth.n
album.n	contrabassoon.n	drummer.n	band.n
include.v	be.v	double.j	bluegill.n
member.n	furneaux.n	electric.j	trombone.n
keyboard.n	5-string.j	rhythm.n	vocals.n
music.n	keyboards.n	instrument.n	saxophone.n
use.v	ampeg.n	tenor.n	fretless.j

Table 2.1: The fifteen highest-weighted within-sentence collocations in Wikipedia for *bass.n* according to four metrics.

from a 2x2 contingency table:

$c(w_1, w_2)$	$c(\overline{w_1}, w_2)$
$c(w_1, \overline{w_2})$	$c(\overline{w_1}, \overline{w_2})$

where w_1 and $\overline{w_1}$ denote the presence and absence of a word in the context, respectively. χ^2 is then calculated over the rows and columns of the table as

$$\sum_{i}^{n} \frac{(O_i - E_i)^2}{E_i},$$

where i denotes a cell in the table and O_i and E_i denote the observed and expected values, computed from the contingency table.

To illustrate the differences between association statistics, Table 2.1 shows the fifteen highest-weighted collocations of *bass.n* measured according to four statistics, using Wikipedia as a corpus. Similar collocations appear in all four lists, e.g., *drum.n*. However, the differences in each list highlight the metrics' biases. Collocations from

raw co-occurrences contain high-frequency lemmas such as *be.v* and *member.n*, which are largely not sense-specific and likely to occur in the highest-weighted collocation lists of many words thereby distorting the structure of the graph (i.e., the majority of collocations with the target word would also be connected to these vertices). While a stop-list could be used to exclude such words, other high-frequency lemmas not on the list would still be likely to appear among the highest-weighted.

Collocations from Dice coefficient, conditional probability, and χ^2 statistics use the lemmas' relative frequencies, which prevents universally-occurring high-frequency lemmas from having high weights. However, both Dice coefficient and conditional probability are still affected by frequency. The denominator of the Dice value normalizes using the sum $c(w_1) + c(w_2)$. Using the sum penalizes collocations of a frequent and infrequent word, even if the infrequent term commonly co-occurs. For example, consider weighting two collocations (w_1, w_2) and (w_1, w_3) where w_1 occurs 100 times, w_2 occurs 9 of its 10 times with w_1 and w_3 occurs 25 of its 50 times with w_1. The Dice coefficient of (w_1, w_2) is 0.16, while the coefficient for (w_1, w_3) is twice the value, 0.33, despite the fact that nearly all (90%) of w_2's occurrences are with w_1. In the context of sense induction, occurrences of word senses are often unevenly distributed throughout a corpus, with a few occurring more frequently than the rest.[6] Because the Dice coefficient's penalizes collocations with infrequent word types, the highest-weighted collocations will be those occurring when the usage has the most frequent sense. Therefore, few collocations with infrequent senses appear among the highest-weighted, which limits the ability of sense induction algorithms to discover these senses. For example, in Table 2.1, the Dice coefficient collocations contain only higher-frequency words for the music senses of *bass.n*, which is the most frequent sense in Wikipedia.

The conditional probability measure exhibits frequency biases for two types of words: (1) infrequent word types that may occur only a few times but only with the

[6]For example, in SemCor, *paper.n* refers to a cellulose material (e.g., printer paper) three times as often as it refers to a newspaper.

target type and (2) high-frequency types that occur in most contexts with the target. The infrequent types are highly sense-associated; for example, the term *pumpkinseed.n* shown in Table 2.1 is a type of fishing lure commonly used to catch bass. However, because the highest-weighted collocations are dominated by these terms, sense induction methods are likely to omit more-common collocations which are necessary for disambiguating (i.e., most contexts will not contain the infrequent terms, thereby making the sense ambiguous to the model). The high-frequency types are often the better source of collocation information from conditional probability, but may also introduce noise from ubiquitous, unrelated word types occurring in the same context, as seen with *be.v* occurring in the 15 highest-weighted collocations in Table 2.1.

The χ^2 statistic mitigates the frequency biases of the Dice coefficient and Conditional Probability measures by using the ratio of the actual co-occurrences and those expected by chance. This ratio allows both infrequent and frequent words to appear among the highest-weighted, which makes the highest-weighted collocations more suitable for identifying less-frequent senses during sense induction. Examining Table 2.1, the χ^2 list contains collocation for the fish and music senses, unlike the Co-occurrence and Dice lists, but does not contain the rare lemmas found in the list for Conditional Probability. Given the potential for bias in the Dice coefficient and conditional probability, we view using the χ^2 statistic as the most robust method for gathering collocations among those in common use.

2.2.2 Building Collocation Graphs for Sense Induction

Graph-based WSI methods rely on having lexical networks that exhibit modular structure where groups of vertices are more connected to each other than the rest of the network. The vertices in these modules correspond to collocations around a particular sense of a word. For example, Figure 2.2 shows the local neighborhood around *voice.n*

with all edges whose $\chi^2 \geq 3000$, calculated using the ukWaC corpus.[7] This neighborhood contains four densely-connected clusters loosely corresponding to senses for *voice.n* as musical instrument (top), an aspect of the human body (right), a person acting as a religious or authoritative spokesperson (bottom right), and a type of communication technology (bottom left). When a collocation network is constructed from the highest-weighted collocations and adding subsequent lesser-weighted collocations, the network naturally exhibits this desirable type of modularity (Véronis, 2004; Biemann and Quasthoff, 2009) and are scale-free, small-world networks (Watts and Strogatz, 1998), where graphs of different sizes have similar structure.

Three methods have been proposed for building graphs for sense induction. First, Dorow and Widdows (2003) propose a method where for a word type, t, the k highest-weighted collocations of t, denoted as C, are added as edges between the word types in the collocation. Then the k highest-weighted collocations for $t' \in C$ are also added as edges, i.e., forming connections between the neighbors as the earlier example in Figure 2.1b. Second, Véronis (2004) uses all collocations in a corpus above a certain weight, but restricts the collocations to being calculated from contexts of t, which constrains the resulting size of the graph. This procedure has the effect that the strength of relationships between collocations are decided based on their occurrences in the context of t, rather than throughout the whole corpus. Third, Navigli and Crisafulli (2010) include all collocations of t with weight above a threshold δ, denoted C, and then include edges from $t' \in C$ if they participate in collocations with weight at least δ as well. In all three methods, once the graph has been constructed, the vertex for t is removed from the graph in order to analyze the connectivity of the neighborhood without the influence of the connectivity of t.

Of these three methods, the latter two have potential biases in them which may create problematic graphs. First, construction process of Véronis (2004) restricts the

[7]The graph was arranged automatically using the organic layout procedure in Cytoscape (Shannon et al., 2003), which favors visualizing densely connected regions closer together.

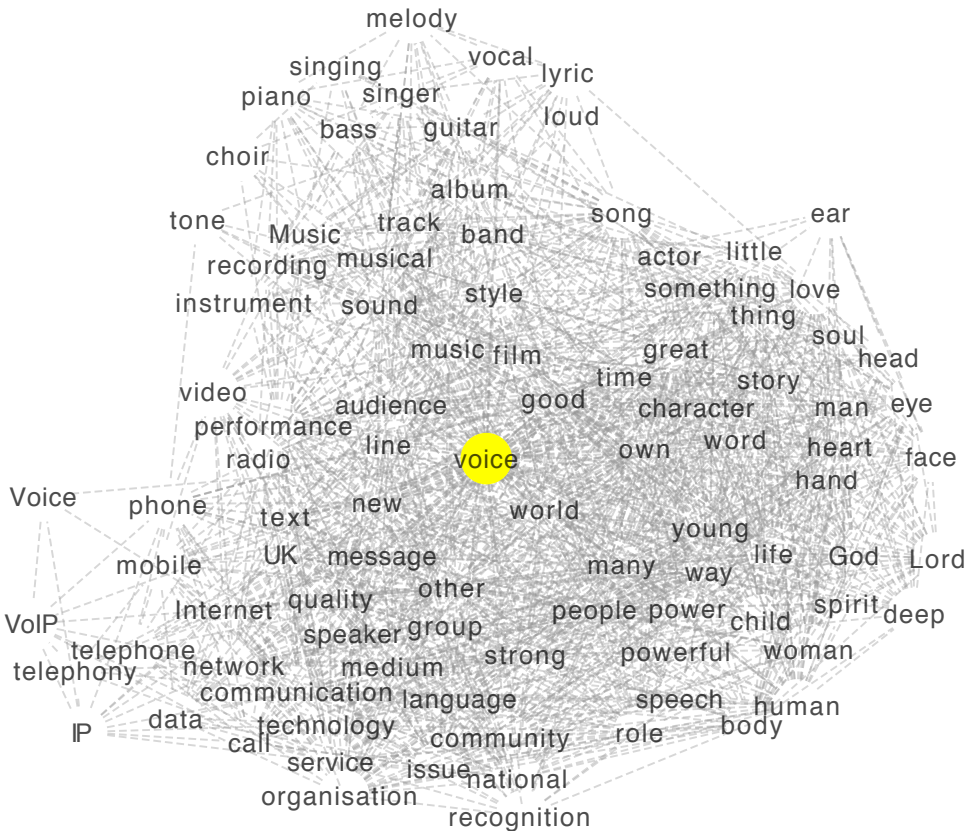


Figure 2.2: An example lexical network showing the neighbors of *voice.n* and all edges with $\chi^2 \geq 3000$ when calculated using the ukWaC corpus. The resulting network shows high connectivity among sense-specific features of *voice.n*.

graphs to only information contained within the context, which in-turn biases the weighting of the collocations. Strongly-associated collocations that are frequent in the target word's contexts become indistinguishable from high-frequency lemmas that are ubiquitous.

Second, both the methods of Véronis (2004) and Navigli and Crisafulli (2010) use a single parameter for all lemmas. Because the magnitude of the weights assigned to each collocation may be affected by the words' frequencies, the single threshold may produce graphs of significantly different sizes. Terms with meaningful collocations that are lesser-weighted may fail to produce a graph at all if the collocations' weights are below the threshold.

The k-neighbors method of Dorow and Widdows (2003) uses the ranks of the collocations, rather than the weights directly. Therefore, the method is not sensitive to changes in the magnitude of the weights and will still produce modular small-world graphs for words regardless of their frequency. We adopt this method for producing the graphs used in this work.

2.3 Learning Word Senses from Graph Connectivity

Sense induction on graphs of collocations is based on the hypothesis that if word w_1 has collocations with words w_2 and w_3 and those words also share a collocation, then both w_2 and w_3 are related to the same *sense* of w_1. This basic relation forms a triadic structure in the graph and when many collocations are related to the same sense and are themselves related, the resulting graph becomes highly connected and in the extreme case is a clique (cf. the earlier example in Figure 2.1). When a word has more than one meaning, the graph contains multiple highly-connected groups of vertices representing related collocations for each sense.

Finding a word's densely-connected sense-specific groups of vertices is complicated by two structural features. First, in most cases, the collocations associated with each sense of a word will not themselves all be related and instead form a mostly-connected group of vertices, rather than a clique. Second, some collocations may be related to multiple senses, partially connecting two groups of vertices.

To find densely-connected groups of vertices in graphs with these features, we draw upon work on the analogous problem on similarly-structured graphs. Specifically, the small-world network seen in the collocation graphs used here mirrors the structure of social networks; and, the two problematic structural features in collocational networks parallel features seen in social networks, where (1) an individual's social network is not generally fully connected and where (2) an individual could be a part of many social groups that are not entirely separate from one another. Research in social networks

has focused on the objective of *community detection*, which tries to find groups of individuals sharing common social bonds purely on the basis of the structure of their social network (Fortunato, 2010). Here, communities are defined as groups of vertices that are more connected with each other than with the rest of the network (Newman and Girvan, 2004).

In our sense induction model, we construct graphs of collocations and then use community detection methods to identify the sense-specific groups of vertices. We propose two sense induction methods using alternate methods of community detection. The first method builds a graph for each term in order to induce its senses and then uses the Fast Modularity algorithm (Clauset et al., 2004) to induce senses. Instead of performing sense induction for one word at a time, the second method performs all-words sense induction by creating a large network of collocations and then using an edge-based community detection method (Ahn et al., 2010) to identify the different senses of all word types in the graph. We refer to the single-word method as SWCD and all-words method as AWCD.

2.3.1 Single-word Sense Induction

Our SWCD algorithm for inducing the senses of a single word consists of three steps. First, a graph for a target term t is built from collocations using the k-neighbors procedure (Sec. 2.2.2) and ranking collocations by their χ^2 association. Note that this graph does not contain t itself. We therefore retain which vertices in the graph were originally connected to t to identify its senses (i.e., to identify the communities into which the collocations of t were placed).

Second, the graph is represented as an adjacency matrix A and Fast Modularity (Clauset et al., 2004) is used to find groups of densely interconnected communities. Fast Modularity is a greedy approximation algorithm that tries to find a partition of the vertices into communities that maximizes network modularity (Newman, 2006),

which is a measure of the degree of interconnectedness of the partition's vertex groups relative to the expected degree in a random network. Network modularity is high when the vertices in each group are highly connected to each other, but not to the rest of the network. Modularity, Q, in our undirected graph of m collocation edges is defined as follows. Let i and j denote vertices represented by the rows and columns of the adjacency matrix A and let $A_{i,j} = 1$ if i and j are connected in the graph and $A_{i,j} = 0$ otherwise. Let k_i denote the degree of the vertex for i. Given two vertices of degrees k_i and k_j, the expected number of edges between them in an undirected graph with the same degree sequence is $\frac{k_i k_j}{2m}$. Then, given a function δ where $\delta(c_i, c_j) = 1$ if the two vertices are in the same community and zero otherwise, we may compute the difference between the number of edges in each community and its expected value to calculate Q:

$$Q = \sum_{i,j} \left[\frac{A_{i,j}}{2m} - \frac{k_i k_j}{(2m)^2} \right] \delta(c_i, c_j) \tag{2.1}$$

Note only vertices in the same community (i.e. $\delta(c_i, c_j) = 1$) contribute to the value of Q and when each vertex is in its own community, $Q = 0$.

Identifying the partition with maximal Q is computationally infeasible as it requires computing Q for all possible partitions of the vertices into non-empty subsets, which is exponential in complexity. To overcome the complexity, Fast Modularity uses a greedy merge procedure to find communities, having complexity $O(md \log n)$ for a graph with m edges and n vertices with average degree d. Specifically, each vertex begins in its own community and then communities are agglomeratively clustered into a dendrogram by merging the two communities whose merger would result in the partition with highest modularity. The dendrogram may then be scanned to find the partition with a locally-maximal Q and cut at that level to produce the final community solution. While many other community detection methods exist (cf. Fortunato (2010) for an extensive review), the low computational complexity of Fast Modularity is highly desirable when dealing with collocational networks containing hundreds of thousands of edges and tens of thousands of vertices.

In our sense induction algorithm's third step, the community output of Fast Modularity is analyzed to find the communities containing neighbors of the target term t. These communities are pruned to remove any with three or fewer vertices and all remaining communities are treated as distinct senses of t. The pruning stage is useful for removing communities that appear from noisy network structure and are likely too small to reliably distinguish a distinct sense.

2.3.2 All-Words Sense Induction

Just as individuals in social networks participate in multiple social groups, words in collocation networks participate in multiple groups of related collocations. In the single-word sense induction procedure, after a graph is built for term t, the term is removed in order to analyze its neighboring structure to identify the senses of t, i.e., the multiple communities in which t participates. Including t would both unnecessarily influence the connectitivity of the graph but also force the Fast Modularity algorithm to place t in only one community. However, an alternate approach is to let t remain in the graph but allow it to participate in multiple communities. To accomplish this approach, we adopt community detection methods that have a similar goal of finding densely connected regions, but relax the constraint that communities must be disjoint, i.e., a vertex may be in more than one community.

Finding overlapping communities enables an alternative to single-word sense induction: A graph may be built from all of the highest-weighted collocations in a corpus – irrespective of any target word – and then an overlapping community detection algorithm may be used to find the senses of all words in the graph on the basis of the communities in which they participate. As Biemann and Quasthoff (2009) showed, collocation graphs containing the k highest-weighted collocations within a corpus are both scale-free and small-world. Thus, more lower-weighted edges may be added to expand the number of word types while still retaining the graph's small-world structure and suitability for use with community detection.

Our all-words sense induction method, AWCD, proceeds in three steps. First, we compute the χ^2 association between all words in a corpus and construct a graph of all associations where $\chi^2 \geq \tau$. As collocation networks are scale-free, the parameter τ is used only to determine the size of the network, with networks produced from different τ having approximately the same structure.

Second, the edge clustering algorithm of Ahn et al. (2010) is used to identify overlapping communities. In the algorithm, edges rather than vertices are evaluated for their participation in community-like structure. Given two edges, $e_{i,j}$ and $e_{i,k}$ their similarity is computed as:

$$sim(e_{i,j}, e_{i,k}) = \frac{n_j \cap n_k}{n_j \cup n_k},$$

where n_i denotes the set containing vertex i and its neighbors. This similarity reflects the percentage of collocations in common with the terms for vertices j and k, independent of those with the term for the shared vertex i. Similarity increases as the two endpoints of the edges share more of their collocations, reflecting the two edges being embedded in a densely inter-connected part of the graph.

In the algorithm, edges are ranked by similarity and then single-link hierarchical agglomerative clustering (Manning et al., 2008) is used to build a dendrogram over merge operations between edge-based clusters. Each cut of the dendrogram produces a partition over edges, which in turn is used to produce overlapping clusters of vertices, also known as a *cover*. To create a cover, each edge partition becomes the set of vertices that are endpoints of its edges.

To derive the final output, each cut of the dendrogram is evaluated according to the quality of its communities. Because network modularity (Eq. 2.1) is defined only for partitions of vertices, not covers, Ahn et al. (2010) propose an alternate method for evaluating the goodness of an overlapping community solution in terms of its connectivity. Let M denote the number of edges in the graph, c denote a specific community, and n_c and m_c be the number of vertices and edges in cluster c, respectively. The density of

the community solution is defined as

$$D = \frac{2}{M} \sum_{c} m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)},$$

The denominator within the summation reflects the percentage of edges present within the community relative to the maximum number possible. In the case of $n_c = 2$, D is defined as zero. Just as with network modularity, D increases as the communities contain vertices that are more connected with each other than with the rest of the network. Ultimately, the dendrogram is cut the level that maximizes D, producing a set of overlapping communities.

In the third step of our algorithm, each word is associated with the set of senses, according to the communities in which it participates. Just as in the single-word method, we remove communities with three or fewer vertices. This filtering step is highly important for the all-words approach, as the edge clustering method has a tendency to put relatively-isolated single edges in their own community (i.e., size two), which artificially inflates the number of senses.

2.3.3 Induced Sense Disambiguation

Each induction method represents a word's senses as a set of related collocations with that sense. To disambiguate a new context according to these senses, we compute the overlap between the lemmatized context and each sense. In the event of ties, we report the sense associated with more collocations under the assumption that this sense is more frequent (as evidenced by having more collocations from occurring a wider variety of contexts) and therefore is more likely to be the meaning.

2.4 Example Results

The senses induced by each method are illustrated by showing (1) features from each of the collocation communities and (2) a keyword in context (KWIC) for usages tagged

with the induced senses. Both methods induced senses using the ukWaC (Baroni et al., 2009), a large web-gathered corpus. Tokens were lemmatized and POS-tagged using the TreeTagger (Schmid, 1994) and then lower-cased to produce the set of word types. Collocation weights were calculated using the χ^2 statistic. As example words, we consider 243 ambiguous words used in prior SemEval WSI tasks (Agirre and Soroa, 2007; Manandhar et al., 2010; Jurgens and Klapaftis, 2013).

The systems were configured as follows. For the all-words method, the graph was created by including the 20,000 most frequent word types (excluding stop words) that were noun, verb, adjective, or adverb. The 50 highest-weighted collocation were added for each of these types. This produced a graph with 964,665 edges and 102,251 vertices. For the single-word method, graphs were created using the k-neighbors method with $k = 1000$. Both community methods are non-parametric, so the number of senses is determined from the graphs themselves.

Because the number of collocations included in each sense community can potentially be in the thousands, we summarize each community using its seven collocations that have highest-weight with the target word.[8] For constructing the KWIC, 10,000 usages were randomly gathered for each target word from the ukWaC, where usages were required to have between 10 and 40 tokens.[9] Usages were then disambiguated with the method described earlier in Section 2.3.3. To better highlight the sense's meaning, each sense's contexts were ranked as follows. Given a context with the set of content words C,[10] let C_s denote the subset of those words associated with collocations for sense s and let $weight(w_i)$ denote the collocation weight (χ^2) for word $w_i \in C_s$. A context is then weighted as $\frac{\sum_{w_i \in C_s} weight(w_i)}{|C|}$. This weighting favors contexts containing strongly associated collocations, normalizing by the total length of the context so that longer contexts are not overly favored. We then show the three highest-ranked contexts for

[8] In the event that a sense is associated with fewer than seven, all collocations are shown.

[9] Because the corpus is web-gathered, this avoided including contexts that were too long or short due to HTML processing errors.

[10] We consider noun, verb, adjective, and adverbs as content words.

different senses of a word in the KWIC.[11]

For the SWCD and AWCD methods, Tables 2.2 and 2.3 show examples of collocations associated with induced senses and below each group the collocations, example usages the sense. Both methods effectively identify fine-grain distinctions in a word's meaning. For example, the SWCD method identifies the distinction between winning an object and winning a competition for *win.v*, between a political campaign and a social movement for *campaign.n*, and between having a darker hue and being devoid of light for *dark.j*. Similarly, the AWCD method identifies distinctions between narcotics and medicine for *drug.n* and between an object that emits warmth and an object that retains warmth for *warm.j*.

Examining the number of collocations associated with each sense reveals a key difference between the methods. The SWCD approach averages 326.1 collocations per sense, while the AWCD method average 9.7 per sense. The smaller number of collocations found in the AWCD method is due to its computational cost; because the all-words method is $O(|E|^2)$ in complexity,[12] only a limited number of collocations may be included for any word, which ultimately results in fewer features per sense. The collocations associated with the induced senses from the AWCD method represent high-precision features that are likely to always select the correct sense when disambiguating; however, due to their sparsity, the method is likely to have reduced discriminative capabilities when contexts contain none of the collocations associated with the senses.

[11]Because of the fixed number of contexts used for building the KWIC, some senses occurred fewer than three times. In such cases, we report all the available contexts.

[12]Specifically, the method is $O(|E|^2)$ in time complexity due to agglomerative clustering and $O(|E|^2)$ in space complexity due to the need for keeping all pair-wise edge comparison in memory in a priority heap. While the time complexity can be overcome just by running the algorithm for sufficiently long periods, the space completely often proved the deciding factor for size of the graph that could be processed. For example, the 1M edge graph used in creating these results required approximately 18GB of memory – even when using space-optimized data structures.

sanctions.n election.n against.n solidarity.n union.n support.v campaigning.n		
During the election	campaign	in May I travelled across the country.
Churchill contested twenty-one election	campaigns	from 1899 to 1959.
The end of the	campaign	was run like we were fighting an election.
launch.v advertising.n awareness.n marketing.n archived.j casi.n uncaged.j		
Gorbachev launched his	campaign	for glasnost (openness) and perestroika.
The modifications proved straightforward and a marketing	campaign	was launched.
A television advertising	campaign	will be launched shortly to support the brand's development.
llaguno.n bolivariana.n outlooks.n post-14.j isitfair.n heats.n anti-stigma.n		
Many of the NCADC 's affiliate	campaigns	are evidence of this fact
[. . .] opportunities for members to participate and support the	campaign.	
I am a member of the Truth About Rape	campaign	and I am writing to complain about this
brown.j light.j pale.j blue.j colour.n black.j hair.n		
They have	dark	brown fur , lighter brown or grey underneath.
They have the ability to change colour from a	dark	brown colour to light green.
Eyes and toe-nails to be	dark	brown or blue.
light.n cloud.n dark.n eye.n shadow.n night.n sky.n		
Like the shining of a torch, or a light in a window, on a	dark	winters night.
Lights come on in a nasty,	dark	room.
I can see light at the end of this	dark	tunnel
wireless.n connect.v ethernet.n networks.n computer.n access.n internet.n		
Will the broadband phone service work over a wireless	network?	
Wireless networking, which will look at providing broadband	network	access using wireless techniques.
It 's easy to add another wireless computer to a Wi-Fi	network.	
neural.j intelligence-g.n reading-inference.n hyperstructure.n human-animal.n purposeful.n efference.n		
Two options for this are neural	networks	and statistically-based heuristic rules.
The difference in the question we asked of the neural	networks	was subtle but important.
[. . .] some experiments with noughts and crosses playing neural	networks.	
rail.n tibus.n support.n bt82.n organisation.n information.n development.n		
He also wants more control of the London rail	network.	
And what about the black hole in Scotland's rail	network,	Fort William-Inverness
Some details of the Iraqi rail	network	on the Come Back Alive website.
match.n game.n win.n victory.n chance.n beat.v season.n		
Except that nobody could	win	the match for us.
When you have five match points you ought to	win.	
The better you know the game, the better your chances of	winning	it.
award.n prise.n championship.n medal.n cup.n competition.n trophy.n		
If so, you could	win	a prize.
Well, she	wins	prizes and things.
To	win	one award is special - but two awards is unique.

Table 2.2: Example senses of four words, where each sense is shown as its highest-weighted collocations and immediately below a KWIC for usages tagged with that sense. Senses were learned using the SWCD WSI method.

anti-inflammatory.j non-steroidal.n antipsychotic.j nonsteroidal.j anti-cancer.j chemotherapy.n aspirin.n		
Avoid NSAIDs (non-steroidal anti-inflammatory	drugs)	if possible.
[. . .] with non-aspirin non-steroidal anti-inflammatory	drugs.	
Ibuprofen belongs to a group of	drugs	called non-steroidal anti-inflammatory drugs (NSAIDs).
decriminalisation.n decriminalised.v criminalisation.n criminalised.v		
Transform A policy think talk arguing for	drug	decriminalisation.
This is the party whose manifesto pledged to ‘maximise	drug	prevention,’ but [. . .]
[. . .] moving cannabis to a Class C	drug	would not amount to decriminalisation or legalisation.
alcoholism.n alcoholic.n alcohol-related.j binge-drinking.n		
[. . .] Sad news regarding alcoholism,	drugs	and incarceration [. . .]
[. . .] all kinds of addictions besides alcoholism and	drug	problems.
Topics include groupies, alcoholism,	drugs,	the glam image [. . .]
confer.v exercisable.j expedient.j paragraph.n inserted.n article.n person.n		
Indeed there are elements of it in the	powers	conferred on the ARA by POCA.
Regulation 5 is made in exercise of the	powers	conferred by s.64(3).
The	powers	in the Act are conferred on the Secretary of State.
electricity.n chp.n hydro.j hydroelectric.j megawatt.n hydro-electric.j gas-fired.j		
The electricity will be fed into the mainland grid via	power	cables.
Do we need to connect up to an electricity supply for	power?	
But its electricity is distributed over all that area , providing light and	power.	
kingly.j endue.v longsuffering.n dishonour.n keepeth.v fleshly.j fount.n		
[. . .] to us who are being saved it is the	power	of God.
[. . .] God’s omniscience and healing	power,	and prayers of repentance and hope.
Babylonian-European astrology, with its emphasis on the	power	of physical bodies on events on earth, [. . .]
plyometric.j fast-twitch.n plyometrics.n one-leg.n		
[. . .] plyometric and strength training to get all the benefits of increased	power	- without putting on additional body mass?
Wherever you look in the world of top-class sport,	power	counts; and one of the best ways of developing this [. . .]
soak.n thermocline.n thermos.n upwelling.n frigid.j airstream.n unheated.j		
The region which separates the	warmer	water of the mixed layer from the colder deep ocean water [. . .]
They tend to shy away from the	warmer	waters of the surface, favouring the thermoclines [. . .]
Treatment [. . .] includes the use of hot soaks,	warm	paraffin applications, heating pads, and joint support devices.
overcoat.n pullover.n raincoat.n poncho.n		
We recommend that you pack some long sleeved jerseys or pullovers, a	warm	jacket, strong footwear and heavy socks.
[. . .] buying her an overcoat to keep	warm?	
[. . .] he had exchanged years before for a blanket to keep her	warm.	
reinstall.v reinstalling.v reinstall.n re-installed.v reinstal.v re-install.n non-viral.j		
You will need to	remove	and reinstall the driver.
If the tests fail to connect,	remove	and reinstall your ADSL modem.
[. . .] and you will need to	remove	and reinstall the driver.
liposuction.n curettage.n ingrowing.j diathermy.n pericardium.n re-growth.n		
Curettage and cautery is another option for	removing	these.
Then a contractor will	remove	the roots, thus preventing re-growth.
[. . .] treatment to reduce pain and swelling and if necessary	remove	the ingrowing nail during a minor surgical procedure.

Table 2.3: Example senses of four words, where each sense is shown as its highest-weighted collocations and immediately below a KWIC for usages tagged with that sense. Senses were learned using the AWCD WSI method.

2.5 Related Work

To perform WSI, two main categories of techniques have emerged: (1) graph-based methods like those discussed in this chapter and (2) distributional methods that represent contexts as feature vectors and then use probabilistic models or clustering methods to discover senses. Below, we summarize both bodies of work by presenting an early example of each and then describing two most recent works.

2.5.1 Graph-based Approaches

Current graph-based methods have applied similar techniques to the two presented in this chapter. Approaches have varied most notably in how they construct their collocation graphs and how clustering has been performed. Dorow and Widdows (2003), Véronis (2004), and Biemann (2006) all use collocation graphs built from co-occurrences, which are biased towards high-frequency terms and are likely to miss less frequent senses. Similarly, Navigli and Crisafulli (2010), Korkontzelos and Manandhar (2010), and Di Marco and Navigli (2012) use the Dice score between words, which is biased towards higher scores for collocations that include rare or ubiquitous words.

Dorow and Widdows (2003) proposed the first approach at inducing senses from graphs. The approach constructs a graph per term using the k-neighbors procedure and co-occurrence frequency to weight neighbors, with the exception that the target word is still included in the graph. Then MCL algorithm (Van Dongen, 2000) is applied to identify clusters in the graph. MCL uses random walks over the graph to identify groups over vertices in which the random walker persists. However, the algorithm is sensitive to two parameters for determining the clusters. Therefore, Dorow and Widdows (2003) use an iterative procedure where MCL is run once to identify the cluster most connected to the target word. Then, that cluster's subgraph is pruned out and MCL is re-run until the remaining vertices do not co-occur with the target work above a threshold number of times. Finally, pruned clusters are merged if their semantic distance is below a

threshold, measuring distance using the technique of Budanitsky and Hirst (2001). The remaining clusters are treated as unique senses. Dorow and Widdows (2003) apply their algorithm only to nouns, using noun co-occurrences, and analyze only the two senses with the largest clusters. While their algorithm is able to avoid manually setting the number of senses, the algorithm is sensitive to parameter tuning and was only evaluated qualitatively.

The method of Dorow and Widdows (2003) is a hybrid approach, incorporating both clustering and pruning. However, later approaches have largely used only one of these methods, either (1) clustering graph's vertices, where clusters denote senses, or (2) pruning the graph's vertices or edges to discover disconnected components that denote senses. We illustrate these strategies with two recent models.

Biemann (2006) proposes inducing senses using the Chinese Whispers (CW) algorithm, a nonparametric graph clustering algorithm. CW is a form of unsupervised label propagation where each vertex is assigned a unique label and then label propagation is run until either convergence or a fixed number of iterations have completed. Label propagation operates by having each node assign itself to the most frequent label among its neighbors' labels. Biemann (2006) constructs the graph in the same manner as Dorow and Widdows (2003) using k-neighbors with co-occurrences, but excludes the target word from the graph. For sense induction, each vertex is initially assigned to its own label (i.e., sense) and then CW is then run over the graph. The resulting clusters of vertices assigned to the same label indicate senses. A major advantage of CW is the low computational cost; the algorithm often requires only a few passes over the vertices to converge, which enables it to be run approximately on large graphs. However, though the method is fast, CW has not been demonstrated to identify groups of densely-connected collocations and the sense induction approach was tested using the artificial setup of (Bordag, 2006) where two word's graphs are merged and the sense induction method must identify which vertices belong to which word.

Navigli and Crisafulli (2010) propose a pruning-based approach for sense induc-

tion, named SquaT. They construct a graph by including all neighboring vertices whose Dice coefficient is above a specified threshold. The neighbors of these neighbors are then included provided that their Dice coefficient is above the same threshold. Navigli and Crisafulli (2010) propose two heuristics for pruning this graph into disconnected components in order to identify the senses. In the most successful heuristic, each edge is scored according to the ratio of the squares in which it participates with its neighbors, relative to the number of possible squares. Here, a square is defined as a closed path of length four. The intuition behind this heuristic is that edges in densely-connected regions of the graph will have end points that are linked together in a square. Edges whose ratio of actual-to-possible squares is below a second threshold are pruned. While the approach performed well for the task of identifying the different meanings of web queries, the method has two potential issues. First, the squares-based pruning method is $O(|E|^4)$ in the worst-case and has an expected run-time of $O((\frac{|E|}{|V|})^4)$ based on the average degree of the vertices. For large graphs, this becomes prohibitively expensive. Second, the method is sensitive to the parameter values, which can result in significantly different graph sizes or even producing no graph at all. Furthermore, tuning these parameters is difficult because of the time required for pruning.

2.5.2 Distributional Approaches

Distributional sense induction approaches were born out of work on distributional, vector-based semantic representations for words. In their most simple form, a vector representation records the frequency of co-occurrence of each word type in a separate vector component (Turney and Pantel, 2010). By computing these vectors from large corpora, words that appear in similar contexts develop similar vector representations. However, because words are polysemous, counting the co-occurrences for all uses of a word produces vector representations that merge the contextual features associated with the word's different senses. For example, the vector for *bass* would be created from the contexts of its fish and musical instrument meanings, resulting in a noisy representation

compared with that of *cello*.

In general, distributional sense induction approaches attempt to separate contexts according to which meaning is present in order to build separate representations for each sense. To perform the separation, contexts are represented as an approach-specific type of vector and compared with each other to identify patterns in usage. Contexts that contain similar features such as their content words are considered to refer to the same sense. However, the information contained in a single context is often very sparse, having tens of content words at most. As a result, two contexts are unlikely to have any overlapping content, despite potentially referring to the same sense of the word. For example, the earlier examples sentences of *pescespada* have no overlap in their content, marked as bold:

- He **grilled** the pescespada **well done**.
- Pescespada goes well with **capers** and **lemon**.
- **Catching** a pescespada **takes** many **hours** in the **open ocean**.

A key challenge for distributional sense induction is to overcome the representational sparsity of contexts in order to identify the semantic regularities.

In the first distributional sense induction approach, Schütze (1992) overcame contextual sparsity by representing a context as a combination of the semantics of its words. Initially, each word type is associated with a vector representation of its co-occurrences with other types.[13] A context is then represented as the average vector (i.e., centroid) its word types' vectors. In essence, a context reflects the average co-occurrences of its word types. Two contexts with no words in common may still have similar vector representations if their content words co-occur with similar word types. To discover senses, Schütze (1992) uses Buckshot clustering (Cutting et al., 1992) to produce a fixed number of clusters. This method of clustering introduces a major limitation, as the number of senses per word must be specified ahead of time.

[13]Note that this vector is itself a noisy combination of the contexts for all senses of the word type.

Two later directions have emerged from this approach. The first direction uses non-parametric clustering to automatically determine the number of senses. For example, Pedersen (2010) and Jurgens and Stevens (2010a) cluster context vectors similar to those used by Schütze (1992) using either K-means with the Gap Statistic (Tibshirani et al., 2000) or Streaming K-means (Shindler et al., 2011), respectively, both of which automatically select the final number of clusters (senses) based on the inter-cluster similarity.

A recent system in this first direction is AI-KU (Baskaya et al., 2013), which represents the context of each target word by using the high probability lexical substitutes according to a statistical language model.[14] A language model is built to identify the relative probabilities of 4-gram sequences and then FastSubs (Yuret, 2012) is applied to identify words that appear in the same position as the target word for each context. For example, one instance of *bass* may have substitutes such as *fish*, while another instance may have *guitar*. Each instance is then represented as 100 substitutes, sampled from the probability distribution of the most-probable 100 substitutes for that instance; these substitutes are transformed into a vector representation, reflecting the sampled frequencies of each. The instance-substitute vectors are then projected into a lower dimensionality using S-CODE (Maron et al., 2010). The final S-CODE based vectors are clustered using k-means. Much like Schütze (1992), AI-KU requires specifying the number of clusters ahead of time, often setting k to a larger than necessary number. However, to determine the number of senses, AI-KU performs a post-processing step to remove clusters that contain only a few instances, which are likely artifacts of forcing each of the k clusters to be non-empty. The remaining clusters are treated as senses of the word.

A second direction of work induces senses using probabilistic topic modeling algorithms such as Latent Dirichlet allocations (LDA) (Blei et al., 2003; Steyvers and

[14] A lexical substitute is a word that may replace another word in context without significantly changing the meaning. For example, in the context "I read the morning paper," "newspaper" is a valid lexical substitute for "paper."

Griffiths, 2007) and Hierarchical Dirichlet Processes (HDP) (Teh et al., 2006). A topic model (TM) is a generative model that describes the content of a collection of documents. A topic is a multinomial distribution over the word types in the collection, which typically take on themes when inferred from a large collection of documents (e.g., animals or music). In turn, documents are modeled as multinomial distributions over topics, reflecting the type of content they contain. In the generative setting, a document with n tokens is created by (1) sampling a distribution over topics, and then (2) for each token, sampling a topic from the document's distribution, and (3) sampling the token itself from the topic's distribution. Importantly, a TM infers the topic multinomials and document-topic distributions automatically based on content of the documents themselves through statistical inference procedures such as Gibbs sampling. A document's topic distribution acts as a higher-order representation in that documents with few words in common may have similar representations as distributions over topics if their contents were generated by the same topics. This higher-order representation enables topic representations to overcome the sparsity of documents themselves.

TM-based WSI approaches learn senses by applying existing topic modeling algorithms (e.g., LDA, HDP) to specialized document collections. In their common use case, a TM is inferred from a general collection of documents in order to discover what commonalities exist in their content (e.g., sports or holiday topics). In contrast, TM-based sense induction methods infer the TM from a document collection consisting only of a word's usages, which discovers commonalities in the contexts in which that word is used. Because the documents are derived only from uses of the target word, the inferred topics reflect the different settings in which the word appears. The underlying assumption behind TM-based WSI methods is that one sense appears per topic and thus each topic corresponds to a different sense. Given a new usage (document), inferring the usage's topic distribution reveals which sense is present.

Earlier WSI models such as Brody and Lapata (2009) and Elshamy et al. (2010) used the LDA topic model, which requires specifying the number of topics (i.e., senses)

prior to inferring the model, just as was required with the WSI approach of Schütze (1992). The more recent approaches of Yao and Van Durme (2011) and Lau et al. (2012) have used HDP models, which automatically infer the number of topics as well as the topics themselves. The approach of Lau et al. (2012) differs in that it includes both content words, positional information (i.e,. the relative offsets of the content words), and syntactic information as features. In an equivalently-tuned setting, Lau et al. (2012) show that this extra information provides a performance improvement over the content-word-only approach of Yao and Van Durme (2011).

2.6 Conclusion

Word sense disambiguation relies on having annotated examples, which can be expensive to produce in large quantities. One alternative to manually annotating instances is to use Word Sense Induction techniques, which automatically learn the senses of a word from how it is used and, in the process, disambiguate those examples as well. This chapter has introduced two new methods for performing sense induction from graphs of collocations, offering two main contributions. First we show that the different methods used for identifying collocations can have a significant effect on the senses contained within the graph; from our analysis we show that using χ^2 offers the least bias and has more potential for containing collocations of multiple senses, irrespective of their frequency in the corpus. Second, we make the connection between the structure of collocation networks and social networks, showing that community detection methods can be used to accurately identify the different senses of a word. We presented two WSI methods based on community detection, the all-words method of Jurgens (2011), AWCD, and a new single-word method, SWCD, showing qualitative results that these methods do identify meaningful senses.

CHAPTER 3

Semi-Supervised Word Sense Induction and Disambiguation Systems

Summary This chapter introduces a new technique for building semi-supervised WSD systems that use induced senses as features for disambiguation, rather than use a context's features directly. This chapter offers three main contributions. First, we review the current method for building semi-supervised WSD systems and then propose a novel method for constructing both individual and ensemble systems, showing that the new method offers superior performance over current state of the art. Second, we propose a new evaluation setting for semi-supervised systems using pseudowords, which enable testing the systems' efficacy on arbitrarily-large amounts of simulated sense-annotated data that closely mimics the linguistic and semantic properties of real sense-annotated data. Third, we use our new evaluation to quantify the performance difference between semi-supervised and fully-supervised WSD systems when limited data is present. We show that our semi-supervised WSD systems offer a statistically significant advantage in cases where only tens to hundreds of annotated instances are available for a word type.

3.1 Introduction

Word Sense Disambiguation identifies the meaning of a word in context. A major limitation for building high-performance WSD systems has been the sparsity of sense-annotated corpora for training supervised systems, and as a result, many unsupervised

or knowledge-based WSD approaches have been proposed (Navigli, 2009). However, an alternate method is to build semi-supervised approaches using Word Sense Induction, often referred to as WSID models (Agirre et al., 2006b). WSID models first use a WSI algorithm to induce a term's senses from its usage in a corpus; then, a WSD system is produced which transforms induced sense annotations into those from a second sense inventory such as WordNet (Fellbaum, 1998). By first using an unsupervised algorithm to learn sense features, WSID systems are potentially able to overcome the knowledge acquisition bottleneck (Gale et al., 1992) by acquiring more information for training a WSD system than is available through sense-annotated data alone. WSI systems automatically group contextual features into senses, which serve as high-level features for the WSID system; a WSID system trained on induced senses is able to map these feature groupings to the second sense inventory, enabling the WSID system to classify new contexts containing features that were not seen in the training data but *were* seen in the data used by the WSI model for learning its senses.

Despite the potential of WSID, the design and evaluation of WSID systems has primarily been done as a part of SemEval tasks focusing on sense induction (Agirre and Soroa, 2007; Manandhar et al., 2010; Jurgens and Klapaftis, 2013). While performance is promising, three important open questions remain. First, in current evaluations, WSID systems have all used the technique of Agirre et al. (2006b) for converting induced sense annotations into those of a reference inventory. However, the performance impact of this process has not been measured, nor have alternative methods been tested. Second, current WSID evaluations have not controlled for the distribution and frequency of the senses in training and test data, which can significantly affect performance and the expected generalizability. Third, despite the potential advantages of WSID for languages with sparse sense-annotated data, no study thus far has directly compared WSID and WSD systems to identify whether one setup should be preferred given a limited number of annotations.

Addressing these questions was previously hindered by the lack of a large sense-

annotated data set. However, we overcome this limitation using the recent resource of Pilehvar and Navigli (2013), which provides a large-scale close approximation of the WordNet sense inventory using pseudowords, which are two or more monosemous lemmas that are replaced throughout a corpus with an artificial token. Deciding which of the lemmas was originally present given an instance of the pseudoword serves as an analogous WSD task.[1] Crucially, in their data set, each polysemous noun in WordNet has a corresponding pseudoword whose pseudosenses closely model the semantic similarity of the original word's senses. Because the data approximates real-world difficulty in disambiguation, the resource enables performing a comprehensive evaluation of WSID on arbitrarily-large amounts of sense-annotated data with direct generalizability to real-world performance.

This chapter has the following four main contributions. First, we provide a comprehensive evaluation setting for WSID that tests systems on millions of instances, two orders of magnitude more than previous evaluations. Our evaluation setting uses high-quality pseudowords that effectively simulate the properties of WordNet senses, which allows us to precisely control the sense distribution of both test and training data. Second, we show that the method for transforming induced senses into WordNet senses has a significant impact on WSID performance, and when using an appropriate method, WSID performance significantly outperforms formerly competitive baselines in multiple tests sets. Third, we demonstrate that combining WSI models into an ensemble WSID provides statistically significant performance improvements. Fourth, in direct comparisons with a state-of-the-art supervised WSD system, we demonstrate that an ensemble WSID system outperforms supervised WSD when fewer than several hundred sense-annotated instances are available, indicating that WSID can indeed overcome the knowledge bottleneck.

[1]For example, the polysemous noun *pic* has two WordNet senses: (1) a motion picture, and (2) a picture. This noun can be approximated by equating each of its senses with a monosemous noun, e.g., *movie* and *photo*, respectively. In the analogous task, a WSD system is asked to determine which of the nouns is present in a context where one originally occurred.

3.2 WSID Systems

A WSID system consists of two key components: a WSI model and a function that converts the model's sense annotations into those of another sense inventory, as formalized in Agirre et al. (2006b). First, a WSI model induces its senses from a *base corpus*. Second, a *training corpus* is labeled using both the induced senses of the WSI model and the senses from a reference inventory. The co-labeled corpus serves as training data for building a classifier that predicts the reference sense label given an induced sense annotation. The present study assesses the performance of WSID systems and therefore we consider a range of sense mapping functions and WSI models, described next.

3.2.1 Sense Mapping Functions

A mapping function is a supervised classifier that given an annotation of one or more induced senses produces a new sense annotation for the instance from a different sense inventory, with the induced senses essentially acting as features. Agirre et al. (2006b) proposed the first mapping function based on matrix multiplication, which has been used in WSID evaluations (Agirre and Soroa, 2007; Manandhar et al., 2010; Jurgens and Klapaftis, 2013). In their setup, a sense co-occurrence matrix M is computed from the training corpus, with columns denoting the n induced senses and rows denoting the m reference senses; the cell $M(i, j)$ records the two senses co-occurrence frequencies. For sense mapping, an induced sense annotation is represented as an n-dimensional vector $\mathbf{v}$ with non-zero values in the columns of the annotated senses. The product $\mathbf{v}M$ produces a m-dimensional vector $\mathbf{v}'$ containing a distribution over reference senses; the sense with the largest corresponding value in $\mathbf{v}'$ is the resulting annotation.

While this mapping function has been used in many prior WSID evaluations, it comes with two weaknesses: (1) all induced senses are considered equally informative for producing the reference sense annotation, and (2) the weights assigned to a sense annotation are not effectively used when an instance's induced labeling has multiple

senses (Jurgens, 2012). Therefore, in constructing WSID systems, we consider six additional supervised learning algorithms for performing the mapping function: Support Vector Machines (SVMs) with both linear and radial basis function (RBF) kernels, Decision Trees based on either entropy or Gini impurity, and naive Bayes classifiers using either Multinomial or Bernoulli distributions. These six classifiers were chosen for two reasons. First, the classifiers have proven effective in a variety prior NLP tasks and second, because the impact of the mapping function has not been tested, using multiple high-quality classifiers will reveal the differences in performance in order to assess how much performance improvement can be attributed to the choice in mapping function.

All six classifiers are trained on feature vectors where each induced sense is a distinct feature and produce a single sense label in the reference inventory. Feature vectors are weighted with the values provided by the WSI models in their annotation, except for SVM, whose instance weights were scaled into [0,1], and for Bernoulli distributed naïve Bayes where all positive values are set to 1 due to its requirement for binary data. Classifiers were implemented using SciKit (Pedregosa et al., 2011).

3.2.2 WSI Models

To assess the impact of the mapping function, we evaluate using five models: our single-word WSI described in Chapter 2, Section 2.3.1, and four other recent WSI algorithm, described earlier in Chapter 2, Section 2.5.[2] The additional models were balanced between those using lexical distributions and those using networks: AI-KU (Baskaya et al., 2013) and HDP (Lau et al., 2012), which use token statistics to induce senses, and Chinese Whispers (Biemann, 2006) and SquaT (Di Marco and Navigli, 2012), which construct graphs to induce senses.

Rather than just reporting the single, highest-weighted sense for a context, we report all detected senses, weighted according to the WSI model's disambiguation step. This

[2]Due to the type of evaluation used (described next in Section 3.3), testing on our all-words WSI model would require months of computation time and is therefore infeasible.

multi-sense labeling potentially captures more information about the context's features and models cases of ambiguity where a mapping function can learn to discriminate which induced sense provides more information predicting the correct sense.

3.2.3 Ensemble WSID Model

Many WSI models -including those used here- exploit different sources of lexical information for inducing senses and thus identify different features for distinguishing those senses. While prior work on WSD has combined complementary WSD systems to improve performance with an ensemble model (Pedersen, 2000; Florian and Yarowsky, 2002; Brody et al., 2006; Søgaard and Johannsen, 2010), no work has pursued an analogous ensemble approach for WSID. We propose a new heterogeneous ensemble WSID system built from the output of all five WSI models. For each instance, the output of the WSI systems is combined and the instance is labeled with the induced senses from all systems. Because WSI models capture different aspects of the context, the ensemble-based system can potentially identify induced senses or combinations thereof that produce a more accurate mapping to the reference sense inventory.

3.3 Experimental Design

To evaluate WSID and WSD systems, all experiments use a common pseudoword disambiguation task. Following, we describe the task and data.

3.3.1 Pseudoword Disambiguation

Pseudowords provide an analogous form of polysemous data for evaluating WSD systems. A pseudoword is made of two or more monosemous lemmas, referred to as its pseudosenses. The occurrences of these pseudosenses in a corpus are replaced with a unique token. In the corresponding disambiguation task, a WSD system must decide

Polysemous noun	Pseudosenses
doubles	badminton, tennis
pic	movie, photo
ca	calcium, california
drawer	desk, treasurer, cartoonist
tapestry	complexity, cloth, rug
headshot	photo, soccer, gunfire

Table 3.1: Examples of pseudowords used in the experiments

which of the pseudosenses was originally present given an occurrence of the token, effectively simulating the traditional sense disambiguation task. Independently proposed by Gale et al. (1992) and Schütze (1992), pseudoword disambiguation fills an important evaluation gap when large amounts of sense-annotated data is unavailable.

However, disambiguation performance on pseudowords is not guaranteed to model the difficulty of disambiguating real words, which may contain related senses (Palmer et al., 2007). To overcome this limitation, Pilehvar and Navigli (2013) present a pseudoword data set where each pseudoword models the sense properties of one of the 15,935 polysemous nouns in WordNet 3.0. Specifically, pseudosenses were selected to closely mimic the semantic similarities of the corresponding senses of the polysemous word. Table 3.1 shows example nouns and their corresponding pseudosenses used in the experiments.

In their analyses, WSD systems were trained on Senseval-3 (Mihalcea et al., 2004) and an analogous pseudosense-annotated data set. The resulting disambiguation performance on the pseudosense-annotated data set was highly correlated with the performance on Senseval-3. Their results indicate that pseudosense-annotated data sets may be constructed to effectively approximate real-world WordNet WSD performance.

3.3.2 Data

All experiments were performed on a subset of 920 pseudowords both for reasons of computational efficiency in building WSI models and for evaluating only on the highest-quality data in the data set of Pilehvar and Navigli (2013). Specifically, their data set includes pseudosenses that are likely introduce noise due to errors from part of speech tagging or taking part in a named entity that was not present in WordNet. Therefore, we exclude pseudosenses where (1) the lemma is also the plural form of another lemma, e.g., spirits, (2) the lemma may be another part of speech, e.g., freezing, and (3) the lemma occurs in fewer than 1,000 contexts in Gigaword when not part of a named entity. To test for the third condition, we used TreeTagger (Schmid, 1994) to identify named entity mentions.

Ultimately, all remaining pseudowords not using these pseudosenses were ranked according to their degree of semantic correspondence with a polysemous WordNet lemma. In descending order of correspondence, we then selected the 920 pseudowords that best approximated the distribution of the number of senses for polysemous nouns in WordNet. Pseudowords in the final data set had between two and twelve pseudosenses with 70% having two senses. Last, because the mapping functions described previously in Section 3.2.1 are parametric, five additional disemous pseudowords were also selected to use in parameter tuning.

3.3.3 Sense Distributions

The sense distribution of a word is often uneven, with one or two senses occurring more frequently than the rest in natural text. The sense distribution can greatly affect WSID performance, with artificially-inflated performances in settings where one sense dominates and all induced senses are mapped to that sense; in such cases, WSID performance does not reflect cases where the sense distribution may vary. While controlling the sense distribution in real-world test data requires a significant number of annotated

instances from which to select, when using pseudowords, the sense distribution may be precisely controlled. Therefore, we consider two extremes in sense distribution in order to better generalize future performance of the models.

In the first distribution, we leverage the correspondence between the pseudowords' senses and WordNet senses to simulate real-world sense distributions based on SemCor (Miller et al., 1993). Specifically, for each pseudoword, we measure the frequencies in SemCor of the senses for the word it models and use their relative frequencies. However, many words are too infrequent to accurately model their expected sense frequencies. Therefore, words with fewer than ten occurrences use the average sense distribution computed from all words having both the same polysemy and at least ten occurrences in SemCor. We refer to the resulting data set as having a SemCor sense distribution.

The second distribution uses a uniform sense distribution. When both training and test data have a uniform sense distribution, WSD systems cannot use the often-effective strategy of defaulting to the most-frequent sense. Though unnatural, the uniform case represents a challenging setting where the performance of WSD models is entirely dependent on their ability to use contextual features without help from a biased sense distribution.

3.4 Experiment 1: Evaluating WSID Mapping

The first experiment measures the impact of the sense mapping function in two ways. First, given the wide-spread use of the Agirre et al. (2006b) mapping function, we assess whether the six alternative functions described in Section 3.2.1 can consistently improve WSID performance.

3.4.1 Experimental Setup

WSID Systems Twenty eight WSID configurations were built for each combinations of the seven sense mapping functions (Sec. 3.2.1) and WSI models (Sec. 3.2.3). WSI models were trained on the same base corpus, ukWaC (Baroni et al., 2009); though we note that models induce senses from the corpus in different ways.

The same WSI parameter values were used for all pseudowords. AI-KU uses the settings for the language model, S-CODE and Fastsubs (Yuret, 2012) algorithms reported in Baskaya et al. (2013). For the k-means algorithm used by AI-KU, k was arbitrarily set at 10, with no further parameter tuning. HDP uses the two parameters to specify the variability of senses in the corpus, γ and α_0, which were set to the same values reported in Lau et al. (2012) and Lau et al. (2013). The SquaT parameters δ and σ were set to 0.00125 and 0.25, respectively, after a limited grid search showed these values produced sufficiently large graphs for inducing the senses of all pseudowords. Our SWCD model built graphs using the k-neighbors procedure using $k = 1000$ after a parameter search on the tuning data (described below) showed this value yielded the best performance.

Seven ensemble WSID systems were built by training each of the mapping functions on the induced sense labelings from all four WSI systems using their default configuration.

Cross-Validation Evaluation Systems were evaluated using five-fold cross validation, with modifications to ensure consistency across distribution types. Initially, a pseudoword is associated with five data partitions, where partitions contain the same number of instances of each pseudosense. The instances of each partition are then filtered to match a desired sense distribution; this filtering process is deterministic so that a partition always has the same instances for a particular distribution across folds. Figure 3.1 visualizes this process. Importantly, this setup ensures that its instances remain

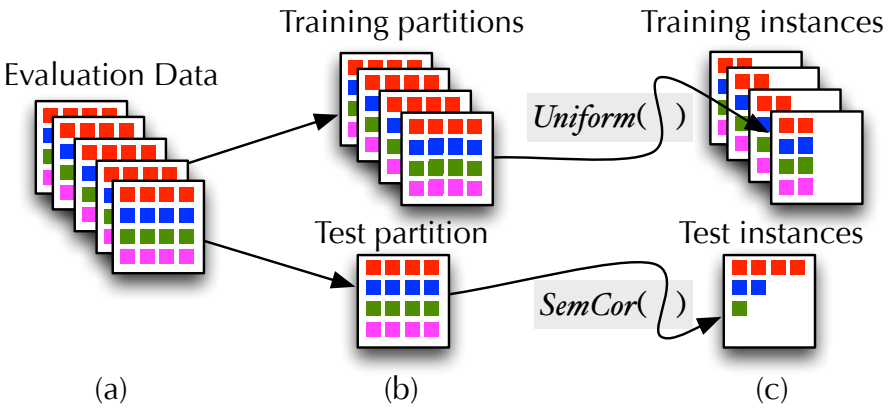


Figure 3.1: A schematic of the cross validation. Evaluation data partitions initially contain an equal number of instances per pseudosense (a), shown as colored boxes. For each fold of validation, four partitions are used for training and one for test (b). The instances from each partition are then sampled according to a distribution (shown in grey boxes), which produces the final training and test instances.

consistent when the partition is used in different folds of validation. We note that in the case of the ensemble WSID system, the underlying WSI models are trained only on the appropriate training data during each fold, ensuring that no test data leaks into the training data.

The reported experiments use either the SemCor or Uniform distributions for both training and test as they are most informative of the expected real-world performance (c.f. Sec. 3.3.3). In the reported experiments, all WSID systems were trained and tested on partitions with the same sense distribution, e.g., training and test partitions having a SemCor sense distribution. However, our evaluation setup is sufficiently general to support evaluating using arbitrary distributions, including using different training and test distributions, as shown in the example in Figure 3.1.

Evaluation Data Evaluation data for the partitions was drawn from the Gigaword corpus (Graff et al., 2003). Instances of pseudosenses were filtered to ensure the cor-

rect part of speech and to remove all occurrences where the pseudosense was part of a named entity. Ultimately each partition in the evaluation data contained 200 instances of all senses, which were filtered according to the desired distribution. For SemCor-distributed data, the most frequent sense has 200 instances, with all other senses having proportional numbers based on their relative sense frequencies. We note that this setup was chosen instead of using a fixed number of instances per partition so as not to bias the results against more polysemous words whose rarer sense would have fewer instances in the fixed-size setting. Because the number of instances varies in the SemCor-distributed data, the corresponding Uniform-distributed data for a pseudoword was balanced to have the same total size.

Two baselines are used with the evaluation data: Random and Most Frequent Sense (MFS). The Random baseline simply picks randomly from among the senses; the MFS baseline selects the most frequent sense of the word, which often performs competitively in skewed distributions such as SemCor and surpassed the performance of many WSID models in previous studies. We note that in the Uniform sense distribution, the MFS and Random baselines are equivalent.

Scoring Systems were evaluated using the standard WSD precision, recall and F1 metrics (Navigli, 2009). Precision measures the percentage of sense assignments provided by a WSID system that are identical to the gold standard. Recall measures the percentage of all instances that are correctly labeled by the system. The F1 is the harmonic mean of precision and recall. When a system labels all instances, precision and recall are equivalent.

Parameter Tuning Five disemous words were used to tune each parametric mapping function. Using a grid search over parameter values, each WSID was scored using an identical five-fold cross-validation process. The parameter values that produced the highest average F1 across all folds were selected for use in the WSID model.

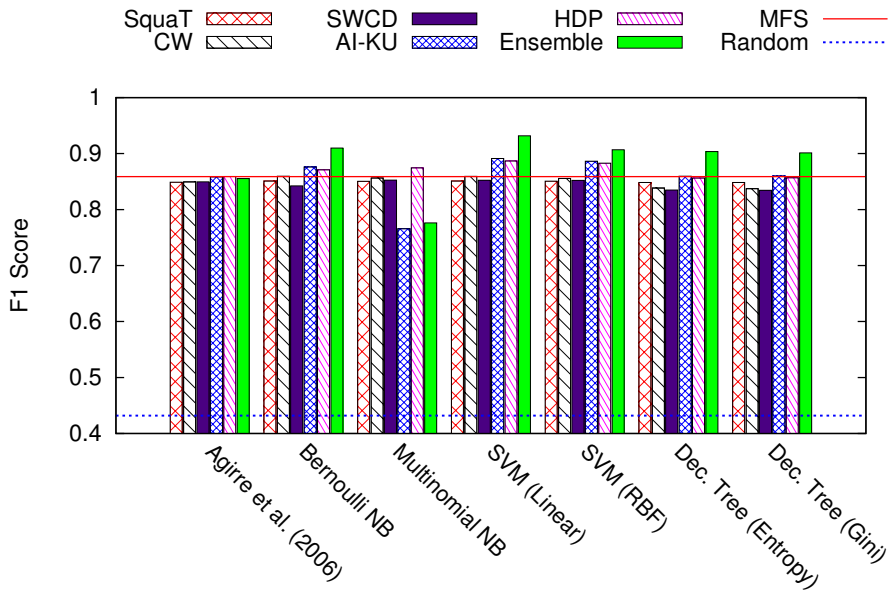


Figure 3.2: Average performance of all WSID systems when training and testing data follow the SemCor sense distribution

3.4.2 Results and Discussion

The WSID system evaluation showed a clear impact from the choice in mapping function. Results for all untuned WSID systems using the SemCor distribution are shown in Figure 3.2. For all WSID systems, tuning the parameters of the mapping function provided little to no performance improvement. We therefore omit tuned results here, but report the scores in the Appendix material. For all WSI models, the commonly-used Agirre et al. (2006b) mapping function does not produce WSID systems that outperform the MFS baseline on average in the SemCor-distributed data. In data with a skewed sense distribution such as SemCor, the mapping function generally maps all instances to the most frequent sense. However, the function does provide higher average performance than the Multinomial naive Bayes WSID systems for SemCor-distributed data because of this tendency to map to the most frequent sense.

The best performance when using a single WSI model comes from SVM with a

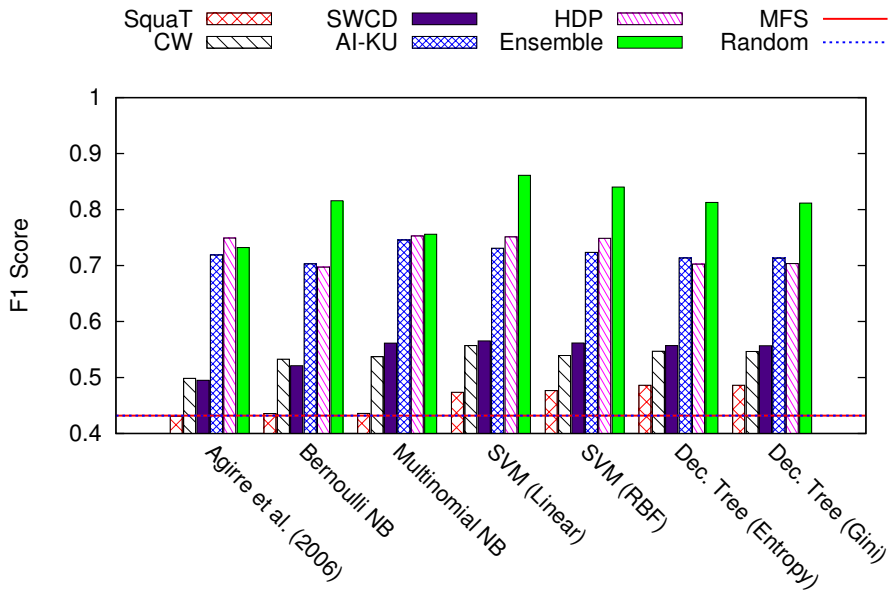


Figure 3.3: Average performance of all WSID systems when training and testing data follow the Uniform sense distribution

linear kernel. Indeed, a WSID system using the AI-KU model and a linear kernel SVM mapping function provides an average 3.24 absolute increase in F1 score over the MFS baseline, which is statistically significant at $p < 0.01$.

Results for the Uniform distribution, shown in Figure 3.3, are consistent with those of the SemCor distribution, though the individual differences of the WSI models are more evident. WSID systems using Agirre et al. (2006b) mapping function perform well on average, but all WSI models enjoy consistently-higher performance when using a SVM mapping function, though the difference between SVM and Agirre et al. (2006b) is not statistically significant for the AI-KU and HDP models.

Results between WSID system show clear distinctions based on the WSI approach. The two distributional methods, HDP and AI-KU, both outperform the graph-based methods for both SemCor- and Uniform-distributed data. Among the graph-based methods, the SWCD approach proposed in this thesis (Ch. 2, Sec. 2.3.1) performs similarly to the state of the art in SemCor-distributed data, but offer a statistically significant

performance improvement over the graph-based approaches at $p < 0.01$. To test the impact of the SWCD approach, we perform an ablation test, removing it from the ensemble and recalculating performance. The SWCD method yields an improvement in F1 of 0.003 and 0.008 for SemCor and Uniform, both of which are significant at $p < 0.01$. Thus, though the SWCD method performs worse than HDP and AI-KU, it still provides a meaningful increase to performance, complementing the methods' WSID abilities.

In nearly all WSID configuration, the ensemble WSID system obtains substantial performance gains. For SemCor-distributed data (Fig. 3.2), the ensemble with a linear kernel SVM produces the highest performance of all WSID configurations, achieving a 7.30 absolute increase in F1 over the MFS and 4.05 increase over the next-closest system (AI-KU). Furthermore, except when using the Agirre et al. (2006b) and Multinomial naive Bayes mapping functions, ensemble WSID systems outperforms all individual WSID systems. When testing and training on a Uniform sense distribution, the ensemble WSID system achieves even more substantial gains over other WSID systems, as shown in Figure 3.3.

The results for both distributions provide two main insights. First, using a linear kernel SVM for sense mapping provides consistently superior WSID performance that is robust to variations in the choice of WSI model. Second, WSID performance can be significantly improved by combining sense annotations from heterogeneous WSI systems with a linear kernel SVM; this improvement was not observed with the currently-used mapping function of Agirre et al. (2006b), which even lowered the performance of the ensemble below that of its best system when tested on Uniform-distributed data (Fig. 3.3). This second finding is key for future work on ensemble WSID methods because it highlights the importance of choosing an appropriate mapping function when combining multiple WSI methods, which had not been identified prior.

To further validate these results, we analyzed the performance of all WSID systems using a linear kernel SVM for different levels of pseudoword polysemy. Due to sparsity, performances for pseudowords with six or more senses were combined and averaged.

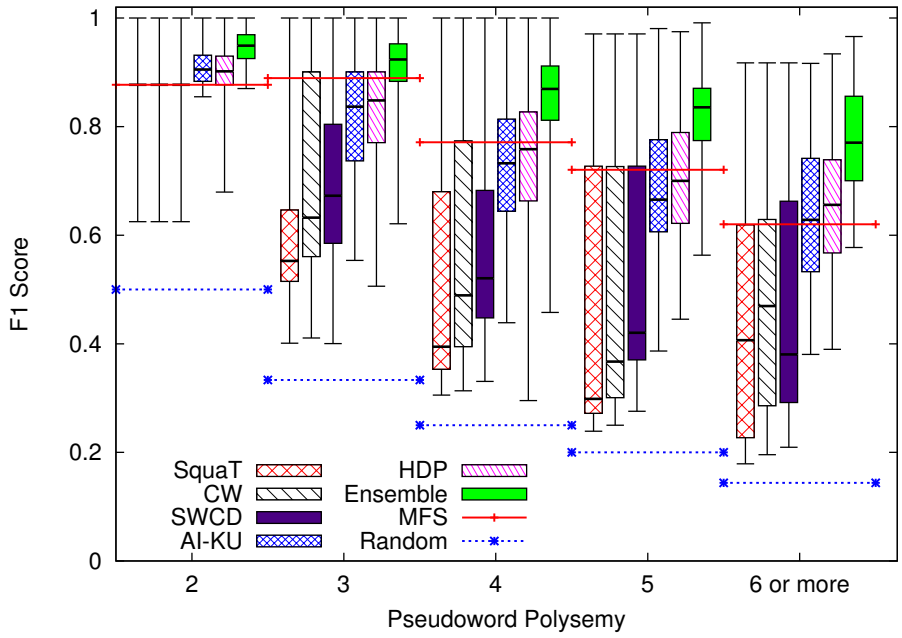


Figure 3.4: Performance of WSID systems using SVM (linear) for different polysemy on SemCor-distributed training and testing data

Figures 3.4 and 3.5 show the performance for SemCor and Uniform sense distributions, respectively, using a box and whisker plot. Whiskers denote the maximum and minimum F1 score of any pseudoword, boxes denote the 1^{st} and 3^{rd} quartiles, and the middle line denotes the median performance. As the baselines' performances change with polysemy, each is plotted as a horizontal line.

As seen in Figures 3.4 and 3.5, the ensemble WSID system offers superior performance across all levels of polysemy. Even in the highly skewed SemCor distribution for disemous words, the ensemble WSID performs better than the MFS baseline for nearly all pseudowords. In the evaluation set of pseudowords, 87% have either two or three senses, for which the ensemble WSID system offered the smallest performance increase over the baseline (though larger than that of any other WSID system). With different data that includes more pseudowords whose polysemy is greater than four, we would expect that the ensemble WSID system would even-further outperform the MFS baseline in average performance.

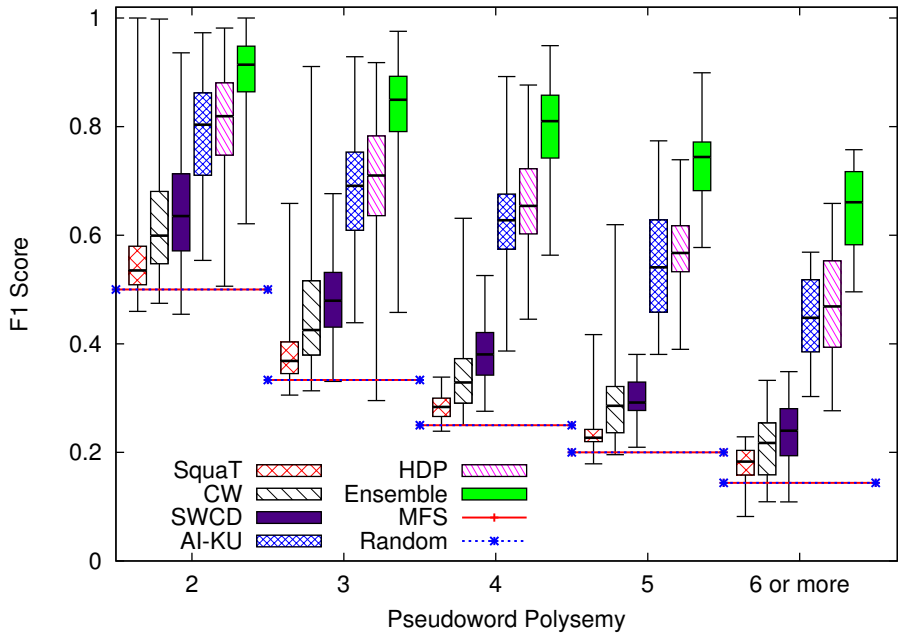


Figure 3.5: Performance of WSID systems using SVM (linear) for different polysemy on Uniform-distributed training and testing data

3.5 Experiment 2: Comparing WSID and Supervised WSD

In resource-constrained languages, limited sense-annotated data may be available. This poses a dilemma where practitioners must decide how to best use the data for building a WSD system: to directly train a WSD system or to use a semi-supervised WSID system. Therefore, in the second experiment, we assess whether WSID can outperform supervised WSD, given identical training data of different sizes.

3.5.1 Experimental Setup

Supervised WSD For a comparison, we use It-Makes-Sense (IMS) (Zhong and Ng, 2010), a publicly-available supervised WSD system with state-of-the-art performance on multiple WSD benchmarks. IMS was trained using its default parameter values.

Training and Test Data IMS training data is generated similar to how instances were allocated for distributions in Experiments 1. For a pseudoword, SemCor-distributed training data is constructed by selecting k instances for its most frequent sense, with other senses assigned a proportional number of instances. Given the SemCor-distributed training data for a specific k with n instances, equivalently-sized Uniform-distributed training data is constructed by including $\frac{n}{m}$ instances for the pseudoword's m senses. Because the number of instances for the most frequent sense in Uniform-distributed data is based on the SemCor-distributed data at k, we use the notation $\hat{k}$ for denoting the equivalently-sized Uniform-distributed data set.

Training and test data were generated from same Gigaword data used in previous experiments, using a five-fold cross validation. Training data was generated for $k = \{10, 25, 50, 75, 100, 150, 200, 250, 500, 750\}$. Both WSID and IMS systems were trained on the k and $\hat{k}$ data sets created from four partitions and then tested on the fifth, *full* partition. In this setting, both WSID and IMS were evaluated on exactly the same test data as in Experiment 1.

WSID Systems WSID systems were constructed using the same procedure used in previous experiments (c.f. Sec. 3.4.1). For simplicity, we report only WSID systems using a linear kernel SVM, as these provided the highest performance. Full performance scores of all other configuration are available in the Supplementary material.

3.5.2 Results and Discussion

The WSID and IMS performances reveal that WSID systems provide significant benefit over IMS when few annotated instances are available. Figures 3.6 and 3.7 show the resulting F1 scores for SemCor- and Uniform-distributed data, with x-axis drawn at log scale. The random baseline (F1=0.432) is omitted from Figure 3.6 for better visual contrast of the remaining systems.

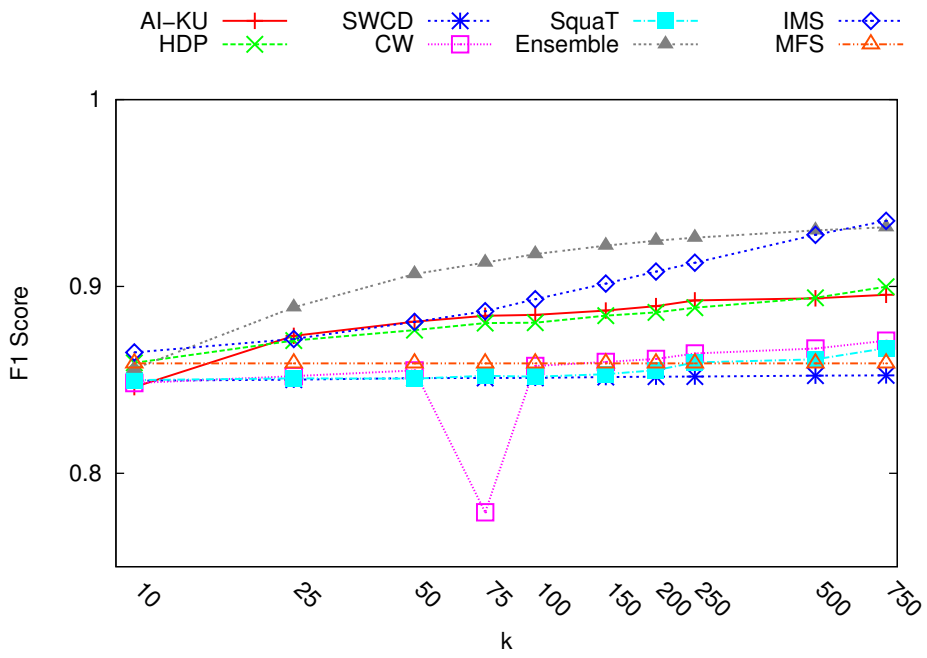


Figure 3.6: Performance of IMS and WSID systems on SemCor-distributed data

As expected, IMS provides superior WSD performance, surpassing all single-system WSID configuration for nearly all values of k and $\hat{k}$. In SemCor-distributed data, IMS outperforms the ensemble WSID at k=10 (significant at p <0.01). However, starting at k=25, the ensemble WSID outperforms IMS until k=750. All ensemble performances 25$\geq$ k $\geq$250 are statistically significant at p <0.01 and the IMS and WSID performances at k=500 are statistically equivalent. In Uniform-distributed data, the ensemble WSID system outperforms IMS until $\hat{k}$=200, where they are statistically equivalent.

Together these results suggest that an ensemble WSID models can offer significant advantages over supervised WSD except when very little or large amounts of sense-annotated data are available. Indeed, all but 97 of the 11,685 polysemous lemmas in SemCor have fewer than 200 instances, which suggests that the ensemble WSID system may offer better performance than existing supervised systems trained only on that corpus. These results indicate that by using the unsupervised features of the WSI models, the ensemble WSID system is able to break the knowledge acquisition bottleneck and acquire more information for disambiguation than available by annotated data alone.

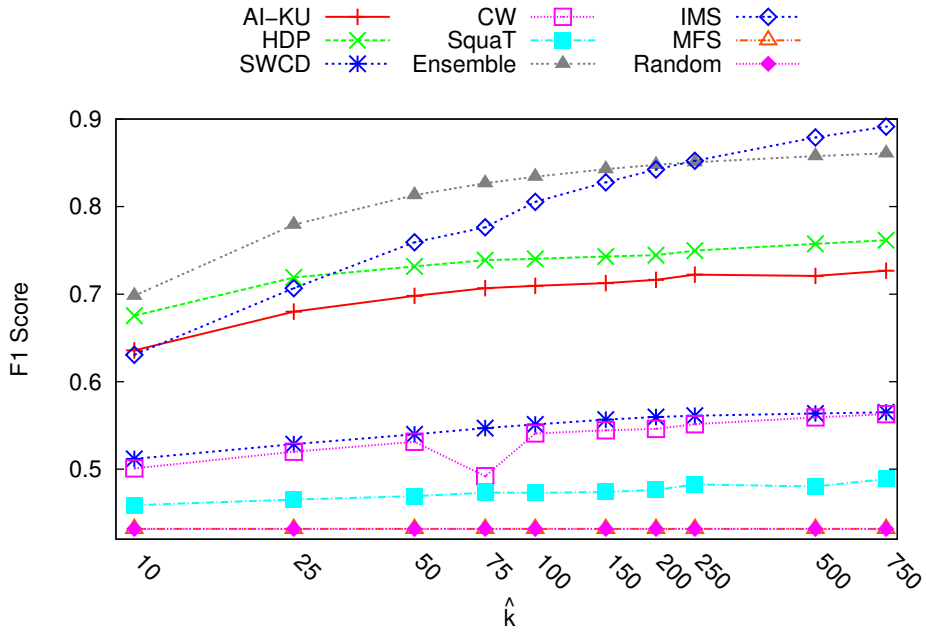


Figure 3.7: Performance of IMS and WSID systems on Uniform-distributed data

Last, we note that increasing the amount of training data consistently improves the performance of IMS, while providing decreasing benefits to WSID models. This contrast highlights the difference in how both systems learn. Training a WSID model on additional sense-annotation data can only improve performance up to the discriminatory power of the underlying WSI systems. Should the training data come from a new domain or context features not see in the WSI systems' training, this new data cannot improve performance. In contrast, providing the same data to a supervised WSD system may enable it to learn new features since the disambiguation system itself can be revised to include such features. Nevertheless, the performance of WSID does depend on having sufficient sense-annotated data to build a correct mapping function, as shown in the large performance improvements between k=10 and k=25 shown in Figures 3.6 and 3.7. Having too few sense-annotated examples prevents the WSID system from learning an accurate correspondence between the induced senses and those of the reference inventory.

3.6 Related Work

The connection between sense induction and disambiguation has been explored by several works. Purandare and Pedersen (2004) and Niu et al. (2005) produce sense induction models and then assign the induced senses directly to reference senses, rather than creating a mapping function that converts induced sense annotations. Both report finding induced senses that closely correspond to existing definitions in reference sense inventories but neither analyze the performance at disambiguating new instances with reference senses, which is the role of WSID systems in this study. As noted in Section 3.2, Agirre et al. (2006b) first formalized the WSID process. In their experiments, a WSID system was built from the HyperLex WSI model (Véronis, 2004) and their mapping function; the resulting system obtained a 0.06 improvement in F1 score over the MFS baseline with default parameters and a further 0.11 improvement over MFS when tuned. Last, in Jurgens (2012), we note the potential for having WSI models annotate items with multiple senses and proposes a modification to the mapping function of Agirre et al. (2006b) to improve performance when multiple induced senses annotation is weighted. In their experiments, WSID systems with this new mapping function were able to outperform a MFS baseline. However, their changes to the mapping function did not address the issue of treating all induced senses equally when performing a mapping.

Several recent works have analyzed senses using pseudowords. Cook and Hirst (2011) simulate the sense properties of lemmas in the Senseval-3 lexical sample task (Mihalcea et al., 2004) in order to model the process by which lemmas acquire new senses; however, pseudowords are analyzed by contextual features rather than using WSD as done in this study. In Jurgens and Stevens (2011), we create a set of disemous pseudowords where the pseudosenses have varying degrees of similarity, measured using their distributional properties in a corpus. Sense induction models are then tested according to their abilities to discriminate pseudosenses at different similarity

levels. However, distributional-based pseudosenses are not monosemous which limits the ability to use them in new corpora with a potentially different sense distributions of the polysemous pseudosenses. Last, the most similar study is that of Pilehvar and Navigli (2013), whose pseudowords we use; however, their analysis focused entirely on supervised WSD, unlike this study which focuses on WSID.

3.7 Conclusion

This chapter presents a comprehensive analysis of WSID systems using a novel evaluation design incorporating 920 pseudowords from the data set of Pilehvar and Navigli (2013), whose pseudosenses closely approximate the properties and disambiguation difficulty of nouns in WordNet 3.0. In our experiments on over a million instances, we provide three contributions. First, we demonstrate that the choice of the mapping function used to convert induced senses can significantly affect WSID and that a linear kernel SVM significantly improves upon the current state of the art (Agirre et al., 2006b). Second, we demonstrate that when using the a linear kernel SVM, joining multiple WSI models into an ensemble WSID system yields large improvements, which were not seen when using prior state of the art. Third, in a direct comparison with a state-of-the-art supervised WSD system (Zhong and Ng, 2010), we demonstrate that an ensemble WSID system offers superior performance over the supervised system using the same training data except when very few or hundreds of annotated instances are available, suggesting that WSID is a viable mechanism for overcoming the knowledge acquisition bottleneck.

CHAPTER 4

Methodologies for Crowdsourcing Word Sense Annotations

Summary An alternate method for overcoming the knowledge acquisition bottleneck is to gather annotations at scale using non-expert annotators. This chapter introduces two new methodologies for gathering word sense annotations using crowdsourcing, presented in Jurgens (2013). Our two methods modify the annotation task to help non-expert annotators resolve their perceived ambiguity in which sense is present. In a series of experiments, we show that by using our novel annotation design with MaxDiff questions, crowdsourced workers achieve inter-annotator agreement rates on par with those seen in expert-annotated corpora.

4.1 Introduction

Sense-annotated corpora are essential resources for developing and evaluating WSD systems. Large-scale sense annotation efforts have used expert lexicographers or trained annotators to produce high-quality annotations (Landes et al., 1998; Hovy et al., 2006; Passonneau et al., 2010). However, the limited supply of experts and trained annotators creates a significant knowledge acquisition bottleneck for building additional sense-annotated corpora (Gale et al., 1992). For example, in the sense-annotation effort for Senseval-1 (Kilgarriff and Palmer, 2000), Krishnamurthy and Nicholls (2000) report an average annotation rate of one annotation per minute; At such a rate, doubling the size of the largest annotated corpus, SemCor (Miller et al., 1993), would require at least 974

total days of effort (2.66 years), assuming two lexicographers each annotating everyday for a standard eight-hour work day.

To overcome the constraints of expert-driven annotation, many researchers have turned to crowdsourcing for gathering annotations. In the crowdsourcing setting, untrained workers complete tasks on sites such as Amazon Mechanical Turk[1] (MTurk) or Crowdflower[2] in exchange for a small amount of pay per task. Because the sites offer large pools of workers, tens of thousands of annotations may be easily gathered in a short amount of time by distributing them across multiple workers.[3] Snow et al. (2008) showed that for many NLP tasks, these untrained workers were able to accurately annotate at the level of experts when their annotations were aggregated. However, research efforts on gathering sense annotations with crowdsourcing have shown mixed results in obtaining high-quality annotations (Snow et al., 2008; Biemann and Nygaard, 2010; Passonneau et al., 2012b; Rumshisky et al., 2012).

The difficulty of sense annotation can be seen through a comparison with another NLP annotation task, part-of-speech (POS) tagging. In POS-tagging, annotators are presented with clear, agreed-upon classification categories (e.g., noun or verb) and all words are labeled using the same set of categories. In contrast to POS-tagging, sense annotation is more difficult due to three factors. First, the sense definitions change for each word, requiring annotators to learn new classification criteria for each (Kilgarriff, 1998). Second, unlike the distinctions between parts of speech, the distinctions between senses may be fuzzy (Palmer et al., 2007), creating ambiguity as to which applies, especially in cases where too little context is available. Third, while in POS-tagging, a word may be used with only a handful of parts of speech, in sense annotation, a word may be highly polysemous and could have tens of senses. This large number of senses increases the difficulty of classification and the time needed per instance.

[1] `http://www.mturk.com`

[2] `http://www.crowdflower.com/`

[3] To help distinguish the expertise of the person annotating, we use the term "worker" to refer to a person participating from a Crowdsourcing platform and "annotator" when the person has received at least minimal training in the task.

Trained sense annotators are able to overcome these issues by benefit of their expertise, though performance at making sense distinctions for highly-polysemous words or for words with related senses is often lower (Palmer et al., 2007; Passonneau et al., 2009). However, crowdsourced workers do not have the benefit of training. Therefore, we consider how to modify the annotation task itself to make it easier for untrained annotators to produce high-quality annotations with their limited experience.

In this chapter, we propose two new methods for sense annotating using crowdsourcing and evaluate our novel methods in comparison to existing methods. Our methods are specifically designed to address two leading problems for producing high-quality annotations: (1) the corresponding decrease in performance as polysemy increases (Fellbaum et al., 1997), and (2) decreased annotator agreement when there is contextual ambiguity about which sense applies. This work builds upon the prior observation that in cases of ambiguity, a usage may be marked with more than one sense to make its ambiguity explicit (Erk and McCarthy, 2009). In prior crowdsourced annotation efforts, workers were required to choose only a single sense. Because of their unfamiliarity with the senses, workers often disagreed on which sense was present (Passonneau et al., 2012b). We hypothesize that if the annotation methodology allows workers to explicitly or implicitly identify *all* the senses they perceive, their resulting annotations with have higher agreement (i.e., by agreement on the ambiguity itself) and thereby provide a clear annotation of what sense or senses are present.

We test our hypothesis by three experiments comparing four different methodologies for sense annotation, two prior approaches: (1) the common single-sense assignment annotation methodology, (2) the Likert-scale annotation methodology of Erk et al. (2009) and two novel approaches: (1) a two-stage process where workers identify potentially applicable sense and then later rate their applicabilities, and (2) a ranked choice based annotation task that uses MaxDiff questions, which removes the need for a sense-applicability rating scale. Each methodology is evaluated using 400 instances evenly divided across eight words.

Our results offer two contributions. First, we show that when asked to annotate using only a single sense, crowdsourced workers attain very low agreement rates, but, when workers can use multiple senses, agreement improves significantly, even when normalized to account for agreement by chance. We find that worker agree on the ambiguous labeling for a sentence and that allowing them to express such ambiguity in their annotations (e.g., by selecting multiple senses) yields high agreement. Second, we demonstrate that our MaxDiff-based annotation methodology produces highly-replicable annotations that can be reproduced by separate sets of workers. Furthermore, the agreement rates of independently-produced MaxDiff-based annotations are on par with those seen in expert-annotated corpora.

This chapter is organized as follows. First, in Section 4.2, we formalize the task of word sense annotation and review prior work both with expert annotators and with crowdsourcing. Second, in Section 4.3 we describe our new sense annotation methodologies and in Section 4.3.6 how their annotation output will be evaluated. Last, in Section 4.4 we perform three experiments and analyses of the methodologies using thousands of crowdsourced annotations.

4.2 Word Sense Annotation

A word sense annotation task is to assign a word's usage to one of that word's senses, given a sense inventory detailing its different meanings. Figure 4.1 shows an example task of annotating a usage of the homonym *bank.n* according to two meanings. Annotators typically work through multiple instances of the same word (Fellbaum et al., 1997; Kilgarriff, 1998; Hovy et al., 2006), which increases annotator familiarity with the word's senses, though an alternative setting of sequentially tagging all words in a context has been proposed by Miller et al. (1998), finding it less tedious and no slower.

Producing an annotated corpus requires at least two annotators to ensure that the annotations are reliable and high-quality. Reliability is measured according to inter-

I dropped my money off at the **bank** today.

Please select the sense of **bank** in the sentence above:

1. sloping land, especially the slope beside a body of water ("they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents")
2. a financial institution that accepts deposits and channels the money into lending activities ("he cashed a check at the bank", "that bank holds the mortgage on my home")

Figure 4.1: An example sense annotation task for the word *bank.n*

annotator agreement (IAA), which has been measured using a variety of statistics such as percentage agreement (Artstein and Poesio, 2008). When annotators have a high-degree of IAA, the annotations are considered high-quality (Navigli, 2009). For example, Kilgarriff (1998) argues that an IAA of at least 0.8 is necessary for considering a sense-annotated corpus for evaluation, based on a IAA ranging in [0,1] where 1 is complete agreement.

Research on sense annotating has focused on three aspects: (1) modifying the sense inventory to clarify sense distinction and reduce annotator confusion, (2) modifying the annotation task to model annotation ambiguity, and (3) gathering annotations from untrained annotators. In this work, we rely on using an established sense inventory, WordNet (Fellbaum, 1998), and therefore do not consider the first aspect. However, we do consider the second and third aspects by adapting the task itself to the crowdsourcing setting.

Following, we review IAA measures used in sense annotation and formally define how it will be calculated in this work (Sec. 4.2.1). Then, we review prior work in modeling uncertainty in sense annotation (Sec. 4.2.2) and work in sense annotation via crowdsourcing (Sec. 4.2.3).

IAA Statistic	Adjusts for Chance	Supports ≥ 2 annotators	Allows Partial Annotations
Percentage Agreement	no	no	no
Scott's π	yes	no	no
Cohen's κ	yes	no	no
Fleiss's κ	yes	yes	no
Krippendorff's α	yes	yes	yes

Table 4.1: A comparison of different IAA statistics commonly used in sense annotation studies

.

4.2.1 Inter-Annotator Agreement

Inter-Annotator Agreement measures the degree to which two or more annotators select the same senses on the same instances. In the most simple case of two annotators who rate all items, the percentage of agreement provides a useful indication of their overall agreement. However, three factors necessitate using more complicated statistics than percentage agreement. First, the distribution of a word's senses in a corpus may be skewed so that only a few senses are likely. This biased distribution can lead to increased agreement by chance if an annotator always selects the most prevalent sense. Therefore, many IAA statistics include an adjustment for chance agreement so that the overall frequency with which an annotator uses each sense is taken into account. Second, more than two annotators may label the same corpus so an IAA statistic should support an arbitrary number of annotators – especially in the case where adjustment for chance needs to be made across all of the annotators' judgments. Third, multiple annotators may each partially label a corpus, producing a corpus where each item annotated by different subsets of annotators (though each having the same number of annotations). This phenomena is especially prevalent in the crowdsourcing setting, where workers may perform as much or as little annotation as they like. Hence, an IAA statistic should ideally handle cases of partial annotations.

They gave the speaker a **warm** welcome.

Please select the sense of **warm** in the sentence above:

1. having or producing a comfortable and agreeable degree of heat or imparting or maintaining heat (“a warm body”, “a warm room” “a warm climate”)
2. psychologically warm; friendly and responsive (“a warm greeting”, “a warm personality”, “warm support”)
3. inducing the impression of warmth; used especially of reds and oranges and yellows when referring to color (“warm reds and yellows and orange”)
4. freshly made or left (“a warm trail”, “the scent is warm”)
5. easily aroused or excited (“a warm temper”)
6. characterized by strong enthusiasm (”warm support”)
7. characterized by liveliness or excitement or disagreement (“a warm debate”)
8. uncomfortable because of possible danger or trouble (“made things warm for the bookies”)
9. of a seeker; near to the object sought (“you’re getting warm”)

Figure 4.2: A sense annotation task for $warm.j$, using its nine senses in WordNet 3.0.

Artstein and Poesio (2008) provide a comprehensive analysis of the most common IAA statistics (summarized in Table 4.1) and suggest the adoption of Krippendorff's α (Krippendorff, 1980) based on its support for the three desirable IAA properties mentioned prior. Krippendorff's α is able to calculate agreement for an arbitrary number of annotators who may each have only partially annotated the corpus. Based on the global distribution of their annotations, Krippendorff's α adjusts for chance, ranging between $[-1, 1]$, where 1 indicates perfect agreement and -1 indicates systematic disagreement. If annotators randomly assigned senses, α has an expected value of zero.

4.2.2 Modeling Annotation Uncertainty

Words with clear, distinct meanings are often straight-forward to annotate (e.g., the homonym $bank.n$ in Figure 4.1). However, many words are less straightforward due to fuzzy distinctions in their meaning, which we illustrate with an example disambiguation task for $warm.j$, shown in Figure 4.2. The meaning of $warm.j$ in example sentence conveys the idea of a friendly and enthusiastic audience. Examining the senses, we observe that senses 2 and 6 both seem to match; indeed, WordNet 3.0 includes the same example use ("warm support") for both senses. Different annotators may agree that the usage is ambiguous but may disagree on which sense best applies in the context and therefore mark different senses. These disagreements in ambiguous cases lower annotator agreement, which is potentially unnecessary if the annotators can agree on the ambiguity itself.

To measure the potential for this ambiguity, Véronis (1998) relaxed the constraint that an annotator must use only a single sense. In the study, six untrained linguistic students rated 600 contexts each for 60 French terms using a common French dictionary to define the word's senses. The number of senses per term varied significantly between 2 and 38 senses term.[4] The six annotators were asked to rate each context as having (1) one sense, (2) multiple senses, or (3) "don't know." Approximately 70% of the instances were marked as having only one sense. However, 30% of noun, 25% of adjective, and 20% of verb instances were marked as exhibiting more than one sense, demonstrating that cases of sense ambiguity are prevalent across all part of speech classes. Nevertheless, in a follow-up analysis, Véronis (1998) notes that the French dictionary itself was responsible for some ambiguity due to the vagueness of its definitions and that better sense applicability guidelines could resolve at least some ambiguity.

In later annotation studies using the well-established WordNet sense inventory, both Langone et al. (2004) and Passonneau et al. (2006) have noted that allowing multiple

[4] Adjectives had the fewest senses on average (2.4), with verbs had the most (5.8), and nouns falling in between (4.6).

senses is necessary in some contexts. However, neither report statistics on how frequently these cases occur, with Langone et al. (2004) noting that precise annotator guidelines still need to be developed.

Erk et al. (2009) propose modeling the ambiguity with weighted sense assignments: When an annotator perceives multiple senses, they rate each sense according to its applicability. The weights serve to explicitly model cases were one sense clearly applies but where alternate-but-unlikely semantic interpretations of the lemma might also be perceived. To test this type of annotation, Erk et al. (2009) had three annotators rate 50 contexts each for eight verbs, providing weighted sense assignments for all of a lemma's senses. They found that while annotators varied in their rating distributions, the annotators had high pair-wise agreement when measured using Spearman's ρ, varying between 0.466-0.540. In a follow-up paper, Erk et al. (2012) report even higher pair-wise agreement on a ratings for ten contexts of 26 different terms, ranging from 0.52-0.72. This pair-wise agreement rivals that of early sense annotation efforts where annotators were constrained to apply only a single sense per instance (Kilgarriff and Rosenzweig, 2000; Snyder and Palmer, 2004; Mihalcea et al., 2004), suggesting that allowing multiple senses can also provide a reliable form of annotation.

Together, these studies suggest that (1) annotation ambiguity is an uncommon but real phenomena for both experts and novices which must be accounted for and (2) allowing annotators to label uses with multiple senses can still provide high levels of agreement.

4.2.3 Crowdsourcing

Given the time and resources required for experts to sense annotate, multiple studies have measured the ability of crowdsourced workers to perform the same task. In their tests on crowdsourcing multiple types of NLP annotation, Snow et al. (2008) had workers sense annotate a part of data from the SemEval-2007 Lexical Sample task (Pradhan

et al., 2007). Ten MTurk workers annotated all 177 instances of *president.n* accord to its three WordNet meanings: (1) executive officer of a business, (2) head of a country, or (3) head of the USA. Worker annotations were then aggregated by using the majority rating as the final annotation. Snow et al. (2008) report perfect accuracy at disambiguating the instances –even finding an incorrect sense annotation in the task's gold standard data set. While encouraging, the study analyzes performance on a single low-polysemy word with clear sense distinctions and therefore is not predictive of future worker performance on highly-polysemous words or those with related senses.

Sense annotation disagreements between crowdsourced workers can occur due to many causes, such as limited fluency in the task's language, poor worker skill at the task, or even adversarial worker who answer randomly. Therefore, two studies have evaluated methods to identify and remove inaccurate workers. Bhardwaj et al. (2010) had either five or six experts annotate 100 occurrences each for three moderately polysemous adjectives (6-11 senses). Fourteen MTurk workers then re-annotated those same 100 instances and the Krippendorff's α was used to measure IAA for each group. Expert agreement was moderate, ranging in 0.49 to 0.67, while worker IAA was significantly lower, near chance level from 0.08 to 0.25. Bhardwaj et al. (2010) then propose an unsupervised method for analyzing annotator agreement patterns, Anveshan, and use it to identify subsets of workers that are have high IAA. While the authors are able to improve IAA, they were only able to find a subset of workers with high IAA for a single word in their test set.

Passonneau et al. (2012b) later reanalyzed the data of Bhardwaj et al. (2010) and trained a supervised GLAD model (Whitehill et al., 2000) with gold standard labels to recognize high-accuracy workers. Although using the GLAD model produced higher-quality aggregated annotations than using simple majority voting, the overall accuracy was not as high as that of expert annotators.

A major limitation of both Anveshan and GLAD is the need to have workers annotate all items in order to calculate the statistics. In the crowdsourced setting, workers

frequently complete only a subset of an job's total work; forcing workers to complete all tasks can serve as a major deterrent to attracting new workers. Indeed, based on payment methods originally used in Bhardwaj et al. (2010), MTurk workers actively advised each other to avoid performing future work for the authors.[5]

Two crowdsourcing studies have achieved good results on disambiguation low-polysemy, coarse-grained senses. Both studies use senses from FrameNet (Ruppenhofer et al., 2006) which describe the word's intended role in the type of context, referred to as a frame. For example, in the Cooking frame, the verb $bake$ could take on three senses: (1) Apply_heat, "She baked the roast for 10 minutes;" (2) Cooking_creation, "He baked his kids some cookies;" or (3) Absorb_heat, "The cake baked for 10 minutes."

Hong and Baker (2011) analyzed worker performance at disambiguating FrameNet senses in 240 sentences each of six lemmas with two to five senses. Ten workers rated each sentence, with the most frequent answer selected as the final annotation. The authors report three main findings. First, workers were able to obtain high accuracy between 73% and 92% (average 85%); however accuracy was increased by excluding 10-30% of the sentences in which fewer than four workers agreed on the same answer. Second, the authors report that in some cases of disagreement either between workers or between workers and the gold-standard, the instance itself was ambiguous and could have been correctly annotation with either sense, which is what workers did. Third, consistent with prior studies involving annotators (Fellbaum et al., 1997; Palmer et al., 2007), they also found that worker accuracy decreased as polysemy increased, with their only five-sense lemma having a disambiguation accuracy of 73% compared with 92% for the three-sense lemmas. However, their sample size is too small to use in generalizing future performance.

Fossati et al. (2013) propose a sense annotation task using the same instances as

[5]See for example `http://mturkforum.com/showthread.php?1244-Stay-away-from-Rebecca-J-Passonneau`.

Hong and Baker (2011), but invert the annotation task structure. Instead of asking workers to select a word's meaning from a list, the task presents two possible senses from different semantic frames (e.g., Cooking and Cause-Motion) and then asks workers to pick which word in the sentence has the meaning, if any. In each case, the potentially-annotated words were restricted to those having two senses. When workers identify the particular meaning with a lemma, the task serves to both disambiguate the word and identify the sentence's semantic frame. Fossati et al. (2013) report an accuracy of 79.2%. While the study shows that workers can accurately identify senses, the generalizability is limited by the two-sense restriction.

Last, we highlight the study of Alonso et al. (2013), which examined the ability of crowdsourced workers to accurately identify cases where more than one sense of a word applied. Specifically, Alonso et al. (2013) examined instances of regular polysemy in nouns where classes of nouns, referred to as dot types, exhibit the same types of sense distinctions (Pustejovsky, 1995). For example, the dot type of Animal/Meat contains words such as $chicken$ that can be used either as an animal ("I fed the chickens") or as food ("I ate my chicken sandwich"). Uses of a dot type may elicit both meanings, as in "I grow my own *onions* and then cook with them," where *onion* is both a plant and food. In their study, Alonso et al. (2013) had five workers annotate 500 items each for five different dot types. Using Krippendorff's α to measure agreement, their results showed that dot type annotation difficulty varied, with α ranging from 0.10 to 0.69. While performance is low for some dot types, they argue that for at least one, the workers were able to produce annotations whose quality was on par with those of experts.

Together, these prior works provide two key insights. First, the studies of Hong and Baker (2011) and Fossati et al. (2013) indicate that high-quality sense annotations can currently be obtained from non-expert workers when the sense distinctions are clear and when polysemy is low. Second, the works of Alonso et al. (2013) and Hong and Baker (2011) indicate that workers can accurately identify word uses with ambiguous

sense annotations. However, no prior study has tested whether the decrease in performance on moderate polysemy words is due to ambiguity and whether performance could be improved by letting annotators mark such instances as ambiguous, which is the hypothesis pursued by this work.

4.3 Crowdsourcing Annotation Methodologies

To compare crowdsourcing methods for sense annotation, we measure the effectiveness of four methodologies when each is used annotate the same data set. Our comparison includes two existing methodologies, single-sense and Likert-ratings, and two new methodologies, Select-and-Rate and MaxDiff. The methodologies produce the same type of sense annotation but differ in how the annotation task is presented to the worker. We first describe how word senses are shown to annotators in all methodologies and then present how each methodology works individually.

4.3.1 Sense Inventory and Descriptions

All annotation tasks use the WordNet 3.0 sense inventory (Fellbaum, 1998). WordNet provides three types of information each sense: (1) a list of other lemmas that having the same meaning, i.e., synonyms, (2) a short definition of the sense's meaning, referred to as a *gloss*, and (3) for many senses, one or more example sentences in which one of the sense's lemmas is used. Table 4.2 shows an example of this information for *add.v*. The annotation methodologies present senses to workers using all three pieces of information.

4.3.2 Single-sense Annotation

Single-sense annotation asks workers to read the context and select the sense that best matches the meaning of a highlighted word in context. Figure 4.3 shows an example

#	Synonyms	Gloss	Usages
1	-	make an addition (to); join or combine or unite with others; increase the quality, quantity, size or scope of	“We added two students to that dorm room”; “She added a personal note to her letter”; “Add insult to injury”; “Add some extra plates to the dinner table”
2	append, supply	state or say further	“‘It doesn’t matter,’ he supplied”
3	lend, impart, bestow, contribute, bring	bestow a quality on	“Her presence lends a certain cachet to the company”; “The music added a lot to the play”; “She brings a special atmosphere to our meetings”; “This adds a light note to the program”
4	add together	make an addition by combining numbers	“Add 27 and 49, please!”
5	total, tot, tot up, sum, sum up, summate, tote up, add together, tally, add up	determine the sum of	“Add all the people in this town to those of the neighboring town”
6	-	constitute an addition	“This paper will add to her reputation”

Table 4.2: The synonyms, glosses, and example usages for *add.v* in WordNet 3.0

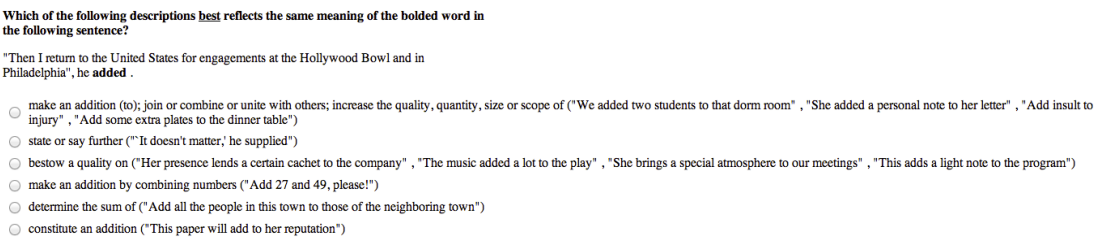


Figure 4.3: A single-sense annotation task for *add.v* on the MTurk platform.

question for *add.v* on the MTurk platform. Single-sense annotation is the most commonly used methodology in expert-based annotation and has been used exclusively in all crowdsourcing based studies. To produce a final sense annotation, multiple workers annotate the same context and then the most frequent sense annotation is selected as the sense label for the context.

Single-sense annotation is a highly-efficient annotation methodology, where each task can be performed quickly (i.e., selecting a single sense) and where a worker only needs to respond to one question in order to annotate the instance. However, its main weakness is the ease with which adversarial workers may click randomly.

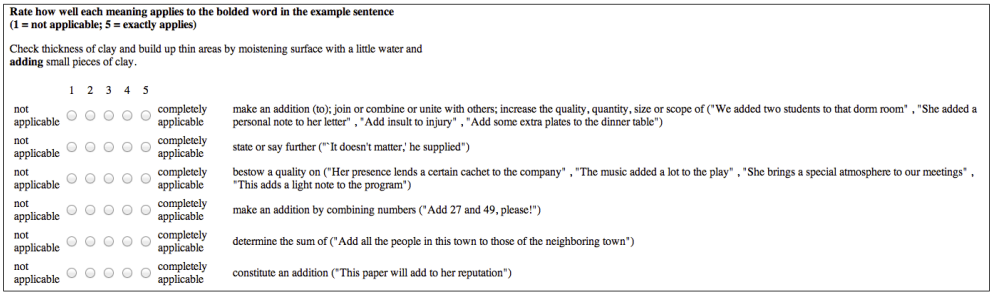


Figure 4.4: An example of a Likert scale sense annotation question on the MTurk platform.

4.3.3 Likert scale Annotation

Rather than require annotators to use a single sense in labeling a usage, Erk et al. (2009) proposed having them rate the applicability of *each* sense for the usage using a Likert scale. A Likert scale is a type of rating scale that models respondent's preferences or selections along a single axis.[6] In the context of sense annotation, the Likert scale was modeled as a five-point a scale from "completely applicable in this context" to "not at all applicable." Figure 4.4 shows a Likert annotation question on the MTurk platform. Likert scale annotation has previously only been tested with annotators but not with crowdsourced workers.

Likert scales provide a direct way to measure worker uncertainty during annotation. Unlike single-sense annotation, workers may select more than one sense and indicate their varying degrees of applicability. We illustrate the potential benefit with two example annotations in Figure 4.5 of earlier example sentence of *warm* (Fig. 4.2). Here, both workers agree on which senses were present, but differ slightly on which sense applies more. In the single-sense setting, workers would be forced to select only one of the senses, leading to disagreement as to what the final sense annotation is. However, with the Likert scale rating, we are able to observe that the two annotations vary only slightly and therefore the final annotation should incorporate both senses. Likert scales

[6] A common occurrence of Likert scales are in questionnaires that ask the respondent to choose a rating on a scale from "strongly agree" to "strongly disagree."

They gave the speaker a **warm** welcome.		
Annotator	Applicability Rating	Sense
Worker-a	5	psychologically warm; friendly and responsive
	4	characterized by strong enthusiasm
Worker-b	4	psychologically warm; friendly and responsive
	5	characterized by strong enthusiasm

Figure 4.5: An example Likert annotation of the context from Figure 4.2 by two workers. All other senses of *warm* are omitted for clarity and were rated "not applicable."

also provide the ability to capture alternate interpretations of workers, where they agree on the most applicable sense but also recognize alternate, less-likely interpretations of the lemma. Such information is lost in the single-sense setting.

To produce the final annotation for a single instance, multiple annotators rate the same instance and their ratings are aggregated. Erk et al. (2009) do not aggregate their annotators ratings, so we evaluate using the mean, median, and mode rating, breaking ties arbitrarily.

Likert scale annotation has two potential disadvantages for sense annotation. First, Likert scales suffer from scale bias, where in the case of sense annotation, workers' mental scales for applicability differ causing them to bias their ratings around one point of the scale. Indeed, Erk et al. (2009) found small scale biases among their three annotators. However, in the crowdsourcing setting, scale bias may potentially be overcome by gathering large numbers of annotations such that the biased ratings are smoothed over. Second, Likert scale annotation increases the task complexity by requiring the worker to answer more questions per instance by providing a rating for each sense. While the total number of tasks is the same as in the single-choice setting, the increased number of questions can slow the completion time.

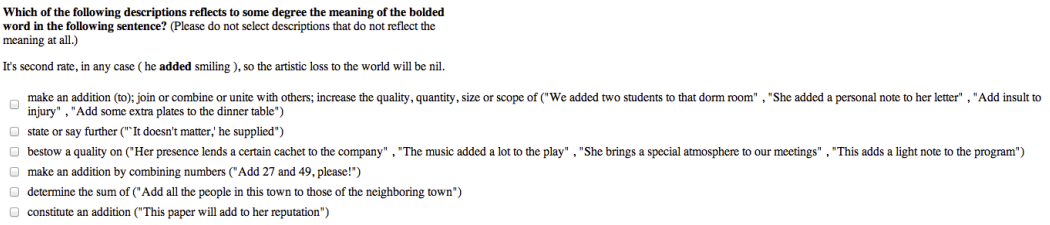
4.3.4 Select and Rate

For the first new methodology, we propose a two-phase process, referred to as Select and Rate. The two stages are designed to first identify which senses *might* apply to a given usage and second, to rate only the potentially-applicable senses. Separating these tasks is motivated by two factors. First, only a few senses are likely to apply to any given use hence, requiring workers to analyze each sense with a Likert scale needlessly increases the cognitive work.

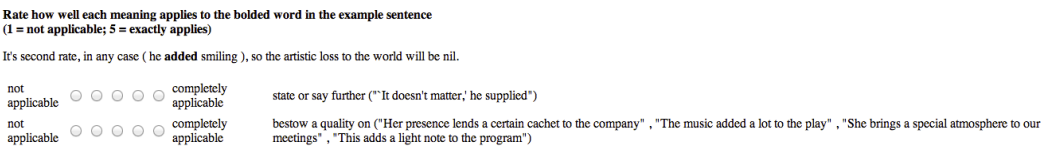
Second, multiple works have shown that crowdsourcing performance on complex tasks such as text summarization can be improved by dividing the task into multiple, conceptually-simpler subtasks, where efforts by one group of workers are used to create new subtasks for other workers to complete (Bernstein et al., 2010; Kittur et al., 2011; Kulkarni et al., 2012). These studies showed cases where higher-quality results were produced from the sequential efforts on the subtasks than from the performing the same operation in a single, more complicated task.

We hypothesize that sense annotation could be divided into two subtasks: (1) eliminating inapplicable senses, and (2) rating applicability of the remaining. By performing these tasks separately, workers can focus in the second subtask only on a smaller set of potentially-applicable senses, leading to higher-quality judgments.

The crowdsourcing workflow is structured as follows. The first subtask, Select, shows all senses of a word and asks workers to make a binary choice as to whether this definition might apply. Although, both this subtask and the Likert-rating methodology requires that workers consider all senses, we hypothesize that making a binary choice will be easier for annotators than picking a point on a Likert scale. Multiple annotators perform the Select task for the same context to produce a rating distribution over which senses might be applicable. Senses whose selection frequency is above a certain threshold are then passed to the second subtask. The second subtask, Rate, operates identically to the Likert rating methodology with the exception only a subset of the



(a) Select task



(b) Rate task

Figure 4.6: A Select and Rate task where MTurk workers chose two senses from the Select task (4.6a) to pass to the Rate task (4.6b).

senses are shown. We refer to the two-stage methodology as Select and Rate, abbreviated as S+R. Figure 4.6a and 4.6b show an example Select task on MTurk for *add.v* where workers have correctly filtered out all but two ambiguous senses, which are then passed to the Rate task.

The design of Select and Rate offers two benefits. First, the Select task can potentially filter the senses to a smaller set using the cognitively less-demanding binary question, thereby allowing annotators to focus their rating faculties only a meaningful subset of senses that might apply. For highly polysemous words with tens of senses, the Select task may even be made easier by performing multiple rounds of Select that each show only subsets of a word's senses and then aggregating those rounds' outputs.

Second, crowdsourcing platforms offer limited visual space for displaying information to workers. The Select task provides a pragmatic way to streamline the task's data so that only a few senses need to be shown on screen for the Rate task. This reduced space can potentially improve annotation throughput and reduce worker frustration at viewing a large bank of Likert scale options.

4.3.5 MaxDiff

Our second novel sense-annotation methodology uses MaxDiff questions to further simplify the annotation task and remove a rating scale altogether. A MaxDiff question is designed to provide information on the respondent's ranking of multiple items with respect to a single quality, e.g., sense applicability (Louviere, 1991).[7] Then, the respondent is shown a partial list of all items and asked select the two items that exhibit the *most* and *least* degree of that quality. In the case of sense annotation, workers are shown a word in context and a subset of the word's senses and then asked to select the sense that most applies and sense that least applies to the given context. Figure 4.7 shows an example MaxDiff question on MTurk for *add.v*, separated into two questions: one asking for the most applicable and one asking for the least applicable.

Importantly, each question shows only a subset of the senses. Multiple subsets are shown in different questions in order to capture the relative ranking between all senses. Showing all senses would fail to capture differences in the ranks of intermediate items (i.e., those not selected as most or least), while showing only two items per question would be equivalent to a Paired Comparison rating, which requires $2^n - 1$ comparisons to rate all n items. Therefore, MaxDiff has two parameters: (1) how many items are shown per question, and (2) how many questions are shown with different subsets. Chrzan and Patterson (2006) recommends of using three to five items per question, which lets the total ranking be evaluated with fewer questions than Paired Comparison without overwhelming the respondent with options. In the example MaxDiff question shown in Figure 4.7, only three of the six senses of *add.v* are shown. For n items, using $1.5n$ unique MaxDiff questions is recommended as a minimum (Sawtooth Software, 2007). We specify the parameter values used in our experiment later in Section 4.4.2 after describing the dataset.

By aggregating the most– and least-applicable responses across multiple MaxDiff

[7]For example, a MaxDiff question may ask a respondent to consider automobiles according to their reliability.

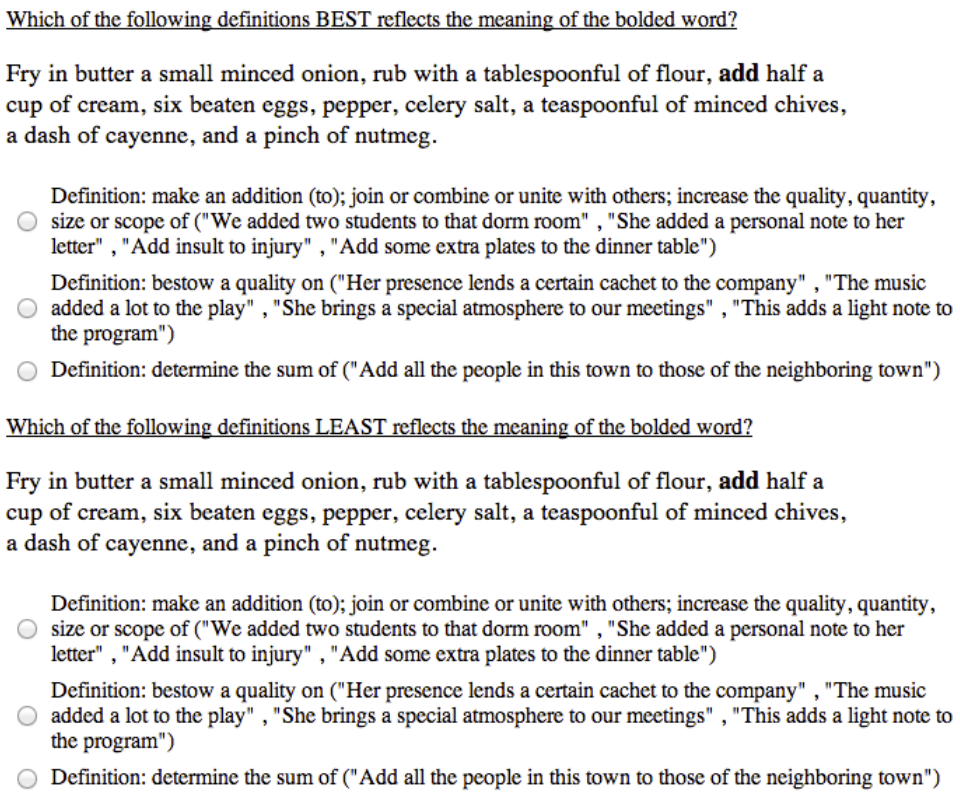


Figure 4.7: MaxDiff questions for the same context, which ask workers to select the most and least applicable senses for the list.

questions for a single context, each sense is assigned a numeric score similar to those produced by scale-based procedures. Sense applicability ratings are produced using a modification of the counting procedure of Orme (2009). Let m_i be the number of times sense s_i was shown and selected as the most applicable and l_i be the number of times it was shown and selected as least applicable. The final rating of each sense is $m_i - l_i$. All negatively-rated sense are treated as completely inapplicable and assigned a score of zero; all positively-rated normalize senses are normalized to have their applicability in (0,1], where one indicates maximal applicability.

Annotating with MaxDiff questions has two advantages. First, because workers answer preference questions, the aggregated ratings are not subject to a scale bias, unlike Likert scale ratings. Second, because only a subset of senses are shown at a single

Sense	Most App. (m_i)	Least App. (l_i)	Final Rating
s_1	10	0	10
s_2	7	0	7
s_3	2	8	-6
s_4	3	10	-7
s_5	1	11	-10

Table 4.3: An example aggregation of MaxDiff responses for five senses where questions showed three senses at time.

time, MaxDiff questions reduce the complexity of the annotation task, much like the reduced-item Rate task. However, MaxDiff does have the disadvantage of requiring more annotations per context: where the previously described annotation methodologies require a constant number of tasks per context (i.e., one for Single-Choice and Likert scale; two for Select and Rate), a MaxDiff annotation requires at least $1.5n$ tasks for words with n senses, which can significantly increase the cost for annotating highly polysemous words (e.g., annotating a single sentence of *hot.j* would require a minimum of 32 tasks).

We note that while MaxDiff is a comparative method, it too allows workers to express their ambiguity in sense annotation. Consider the example aggregate MaxDiff annotation shown in Table 4.3 where MaxDiff questions showed three senses per question. Here, workers have found that senses s_1 and s_2 to be the most applicable when shown in questions. In cases where both s_1 and s_2 were shown together, s_1 was picked as most-applicable more often, giving it a higher final rating. Note that when only the three inapplicable senses were shown in questions, one of these senses must receive a most-applicable rating. However, in the remaining questions where an applicable sense was present, the inapplicable senses were frequently selected as least-applicable, which yields a negative aggregated rating and a final sense annotation of inapplicable. By varying the items shown in each question, each applicable sense can ultimately receive most-applicable ratings and a positive aggregate score, while inapplicable items

become consistently rated as least-applicable. This process allows workers to express their ambiguity without needing to rate each sense along a scale.

4.3.6 Calculating Inter-annotator Agreement

Krippendorff's α is used to calculate IAA for each methodology according to its respective level of measurement, which refers to the scale used to represent and compare responses (Stevens, 1946). In our annotation task, The Single Choice, Select, and MaxDiff tasks use the nominal level of measurement, where responses denote names that are not comparable to one another. In contrast, the Likert scale and Rate tasks use the ordinal level of measurement, where applicability ratings reflect rankings along a value scale. When aggregating Likert scale ratings by taking the mean value, we interpret the value according the interval level of measurement, where the value reflects a rating (rather than a rank) along a defined scale. Krippendorff's α may be computed for all three of these levels, which allows easier interpretation and comparison when looking at agreement rates from different types of annotated data.

Annotation tasks are modeled as follows for computing α. A Single-sense task is considered as a single item with n choices, one for each sense. A Select task is considered as n items, one for each sense; each item has two choices denoting whether its corresponding sense was selected. A Likert scale task has n items, one for each sense; each item has five choices, one for each of five ranks on the rating scale. The Rate task is identical to Likert, but will have between one and n items, depending on the output of the Select task. Finally, a MaxDiff task is considered as two items, one for selecting the most applicable sense and one for selecting the least applicable; each item has k choices, depending on the number of options shown in a MaxDiff question.

4.4 Experiments

We perform three experiments to measure the quality of fine-grained sense annotations produced by crowdsourcing using each of the four described annotation methodologies. The first experiment tests the agreement of the MTurk workers. While crowdsourcing workers are not expected to have a high agreement, this first analysis is designed to reveal which of the methodologies yields higher initial agreement. Crowdsourcing is expected to perform best when worker responses are aggregated (Surowiecki, 2005); therefore, the second experiment measures the quality of aggregated annotations based on their agreement with a reference annotation. The third experiment performs a replicability test to measure the reproducibility of the same annotations from an entirely different group of workers.

4.4.1 Data set

To compare with prior work, experiments were done using the contexts in the Graded Word Sense (GWS) data set of Erk et al. (2009). In their experiments, Erk et al. (2009) had three annotator rate 50 contexts each of eight words using the Likert-rating methodology (Sec. 4.3.3). The eight words, shown in Table 4.4, are of moderate polysemy with 4–7 senses. The words' contexts were drawn evenly from the SemCor (Miller et al., 1993) and SENSEVAL-3 lexical substitution (Mihalcea et al., 2004) corpora. For all contexts, the GWS annotators rated the applicability of all WordNet 3.0 senses, which we also use.

Annotation was done in-person (not through crowdsourcing) and annotators were free to go back and revise their ratings. Annotators were all native English speakers. For simplicity, we refer to them as the GWS annotators.

Term	Part of Speech	Number of senses
add	verb	6
ask	verb	7
win	verb	4
argument	noun	7
interest	noun	7
paper	noun	7
different	adjective	5
important	adjective	5

Table 4.4: The eight terms from the GWS dataset (Erk et al., 2009) used in the crowdsourcing experiments

4.4.2 Crowdsourcing Setup

Crowdsourcing was performed using the Mechanical Turk platform. We first describe the design of the task itself and then describe MTurk platform-specific details.

Each annotation methodology was tested using a methodology-specific MTurk task design. Tasks contain three elements: (1) an IRB header indicating that the worker consents to participating the study,[8] (2) general instructions, shown in Table 4.5, asking the worker to complete all questions and to not participate if they cannot understand the task or meaning of the words, and (3) methodology-specific instructions describing the details of what a worker was to do.

The MTurk platform permits works to complete as many tasks as they please. Therefore, to increase the familiarity with the annotation task, each task included four annotation questions. To further increase familiarity, all questions showed contexts for the same lemma with senses in the same order. This batching also served as a deterrent to adversarial workers that seek to randomly answer a single question and move on. We note that although workers were free to complete only a single task, many completed multiple tasks and a few workers even completed tasks from multiple annotation

[8] All work was submitted and approved under IRB#11-003218.

This study will be used to better understand how readers interpret the meaning of a word based on its context. Your input is much appreciated. If any of the questions in a HIT are unanswered, or some ratings are left blank, then the assignment is no longer useful to us and we will be unable to pay for the assignment. HITs that contain randomly clicked ratings will also not be approved. Skip a HIT if you do not know the meanings of the words. Attempt HITs only if you are a native speaker of English or very fluent in English. Certain check questions will be used to make sure your input is responsible and reasonable. HITs that fail these tests will be rejected. If you fail too many check questions, then it will be assumed that you are not following instructions above, and ALL of your HITs will be rejected. In this HIT will be presented five questions. In the first question, you will be presented with a bold-faced word and a list of possible definitions of that word. Your task is to select which of the definitions is a correct definition of that word. Only one definition should be correct; if you feel multiple definitions could be correct, select the definition you would most commonly use with the word.

Table 4.5: General instructions shown for all annotation methodologies at the beginning of each task on the MTurk platform, referred to as a HIT in the instructions.

methodologies.

For all methodologies, each task was completed by ten MTurk workers. For Select and Rate, senses were passed from Select to Rate if they received at least three votes. For the MaxDiff methodology, questions showed three senses at a time for terms with six or fewer senses and four senses for those with seven senses, in line with the recommendations of Chrzan and Patterson (2006). Each contexts was rated with $3n$ tasks per context where n is the number of senses of the target word. MaxDiff tasks used randomized permutations of the senses, ensuring that each sense appeared at least once. Due to resource limitations, we omitted the evaluation of *argument.n* for MaxDiff.

Our task setup on the MTurk platform includes three best-practices for conducting research with MTurk. First, as Ross et al. (2010) and Mason and Suri (2012) note, MTurk workers come from a wide-variety of demographics, which can lead to incon-

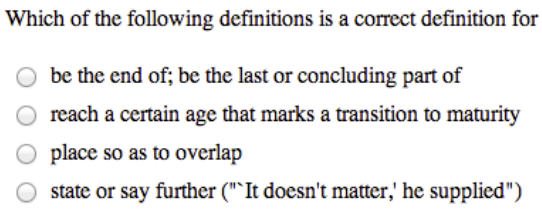


Figure 4.8: An example check question that preceded all annotation tasks, where the Turker must select a correct definition for the bolded word amongst the four options.

sistent results, especially when language fluency is not necessary to submit a response, which is the case for the multiple choice tasks used in our study. Therefore, to ensure fluency in English, we adopt the strategy of Jurgens et al. (2012) and prefaced each HIT with a simple test question that asked the worker to pick out a definition of the target word from a list of four options. The incorrect options were selected so that they would be nonsensical for anyone familiar with the target word. Figure 4.8 shows an example check question for *add.v*. Additionally, we rejected all work where more than one option was missing a rating. Approximately 15-25% of the task submissions were rejected by these criteria (depending on the methodology), underscoring the importance of filtering.

Second, Zhu and Carterette (2010) notes that adversarial workers will frequently resort to patterns when providing multiple choice input, e.g., repeated rating items using a score sequence 1-2-3-4, or rating series of items with a constant score. Therefore, we test for multiple types of patterns in the input, which resulting in rejecting 5-10% of the input per lemma.

Third, in related studies Bachrach et al. (2012) and Kosinski et al. (2012) analyzed the relationship between the payment per task and the capabilities for the responding MTurk workers on a Raven's Progressive Matrices (Raven, 1936), which are non-verbal question for measuring cognitive capacity. Both studies showed that the highest performance was obtained using a payment near 0.05 USD; higher payment resulted in attracting scammers, while lower payment disinterested higher-quality MTurk workers.

	add.v	ask.v	win.v	argument.n	interest.n	paper.n	different.a	important.a
Single-sense	0.332	0.332	0.030	0.139	0.097	0.174	0.007	0.052
Erk et al. (2009)	0.470	0.354	0.072	0.497	0.320	0.403	0.212	0.466
MTurk Likert	0.336	0.212	0.129	0.250	0.209	0.522	0.030	0.240
MTurk Select	0.309	0.127	0.179	0.192	0.164	0.449	0.024	0.111
MTurk Rate	0.204	0.076	0.026	0.005	0.081	0.108	0.005	0.116
MTurk MaxDiff	0.493	0.353	0.295	-	0.349	0.391	0.220	0.511

Table 4.6: IAA per word for the different sets of annotators. The top most methodology restricted annotators to a single sense choice, while the middle group all allowed multiple choices.

Therefore, we adopt the same payment range, paying $0.05USD for the Single-sense, Select, Rate, and Likert tasks. Because we intentionally use more MaxDiff tasks than is necessary, the size of the task necessitated paying less due to financial constraints; payment was ultimately set at $0.03USD, though hourly payment rates were similar due to the higher throughput of the MaxDiff tasks.

In total, 730 workers participated in the experiments and provided 41,280 accepted responses to annotation questions on the MTurk platform.

4.4.3 Experiment 1: Annotator Agreement

The first experiment evaluates each methodology by the workers' IAA. The agreement rates of the workers provides a measure of how easy the task is for the workers. However, we note that unlike the IAA of experts, worker IAA should not be considered a final assessment of the annotation quality, as multiple workers' responses are combined together to produce the final sense annotation. Nevertheless, at least moderate agreement is needed to demonstrate that the workers are not answering randomly.

Results Table 4.6 reports the IAA for the workers participating in each annotation methodology and the IAA for the GWS annotators on the same data set. Two findings were observed. The first finding is that allowing workers to select more than a single rating does improve IAA, which confirms our hypothesis. Consistent with other work on crowdsourcing fine-grained sense annotations (Passonneau et al., 2012b), workers had low IAA when using the single-sense methodology. The Single-sense α values ranged in $[0.007, 0.332]$, with only *add.v* and *ask.v* having an α above 0.1. Unlike prior work, we did not observe any correlation between the degree of polysemy and decreased IAA in the Single-sense annotation setting, which we attribute to worker difficulty in choosing between the related senses of the adjectives and $win.v$.

In contrast to the IAA of Single-sense workers, IAA improved when workers were allowed to explicitly select multiple senses, as in the Likert scale and Select methodologies. Furthermore, Likert scale IAA approaches (and for two words exceeds) the IAA of the GWS annotators who used the same methodology during an in-person study (Erk et al., 2009). This result is encouraging for two reasons. First, crowdsourcing is a highly adversarial annotation environment compared with in-person studies. Second, 170 unique workers participated in the Likert rating annotation tasks and completed on average 28 tasks, compared with the three GWS annotators who were able to build expertise across all 400 instances and who could also revise their prior annotations to improve consistency. The close IAA rates of the workers and GWS annotators demonstrate that crowdsourced workers have similar capablities as in-person annotators on the same task.

The second finding of the experiment is that the choice of annotation methodology significantly impacts IAA. While both the Likert and S+R tasks have lower IAA than the GWS annotators do, the MaxDiff tasks achieve higher IAA for almost all words. We hypothesize that comparing senses for applicability is an easier task for the untrained worker, rather than having to construct a mental scale of what constitutes the applicability of each sense.

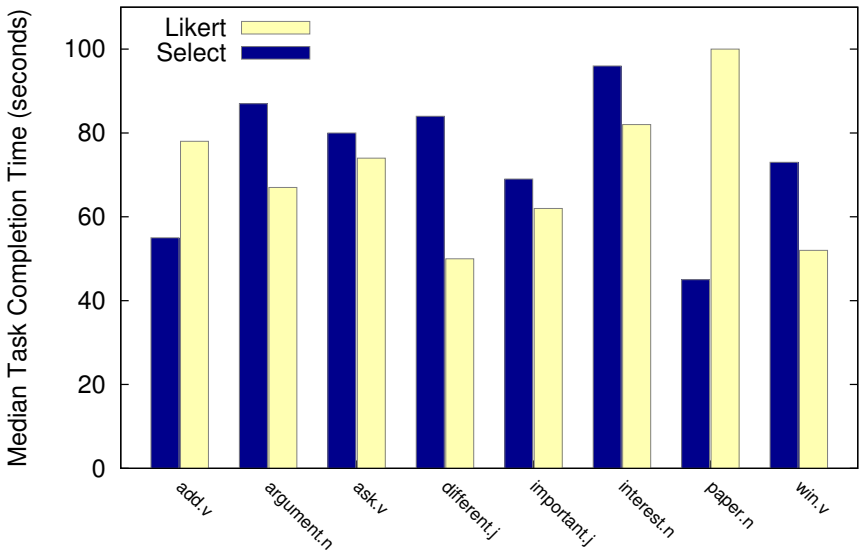
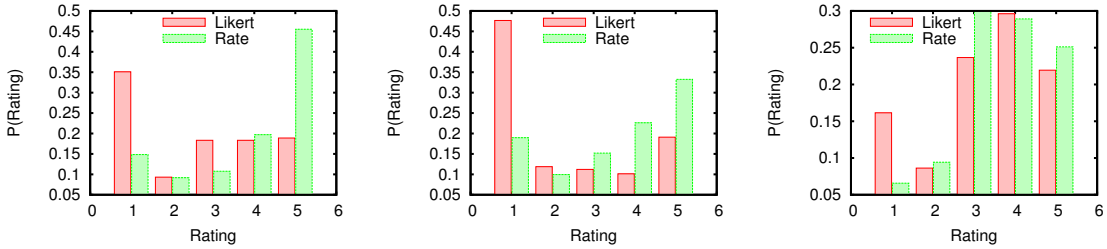


Figure 4.9: Median completion time for one task (four instances) with the Likert and Select formats.

Surprisingly, the low IAA of the binary Select task suggests that it was more difficult than the rating-based Likert-based setting. We hypothesize that Turkers were divided over the choice in what constitutes the cutoff for applicability in the Select task, whereas in the Likert task, questionably applicable senses could be assigned 1 or 2 without a large penalty to IAA. As a way of estimating the task difficulty, we measured the time taken to complete one HIT (four annotations) for both the Likert and Select ratings. Due to the uncontrolled nature of the online setting[9] we compare the median task completion times as an estimate of difficulty instead of the mean times, which are more affected by outlier completion times.

Figure 4.9 shows the median completion times for all approved annotations, revealing that the median duration of the Select methodology is often slightly longer than that of the Likert task, despite Select having a simpler question format. The exception to

[9]In the MTurk platform, Turkers are free to take breaks during the task itself (e.g., making a phone call, surfing the web) provided that their completion time is within the maximum allowed, which was set at the default 30 minutes.



(a) add.v (b) paper.n (c) different.a

Figure 4.10: The probability of each rating across all instances for the Likert and Rate task setups.

this is annotations for "paper," for which the median time was 50 seconds faster with Select and had the highest IAA among all words for Select. As a whole, the similarity of annotation times suggests that a binary decision in applicability is just as cognitively demanding as rating the applicability itself.

Last, the Rate task has the lowest IAA, despite its similarity to the Likert task. One potential explanation would be that because the Select subtask filters out all low-applicability senses, annotators could have used very similar ratings for the remaining senses and potentially had a lower α due to its correction for chance (i.e., when nearly all ratings are the same, agreement is primarily due to chance). However, an inspection of the ratings revealed no such trend. Figure 4.10 shows a comparison of the rating distributions for the Rate and Likert on three words, revelaing that Turkers used the whole rating scale in the Rate task. Furthermore, the use of the whole rating scale indicates that some senses that were above the Select threshold were still considered inapplicable.

4.4.4 Experiment 2: Aggregating Crowdsourced Annotations

The benefit of crowdsourcing is that while individual workers may make mistakes, the combination of their responses produces an answer much closer to the correct answer (Surowiecki, 2005). Therefore, in the second experiment, we evaluate aggregated

	add.v	ask.v	win.v	argument.n	interest.n	paper.n	different.j	important.j	*avg.*
Likert Mean	0.462	0.339	0.102	0.426	0.322	0.518	0.148	0.460	0.347
Likert Median	0.470	0.332	0.093	0.430	0.339	0.535	0.136	0.477	0.351
Likert Mode	0.500	0.369	0.083	0.445	0.388	0.518	0.124	0.516	0.369
S+R Mean	0.474	0.383	0.130	0.487	0.385	0.477	0.094	0.410	0.355
S+R Median	0.473	0.394	0.149	0.497	0.390	0.497	0.103	0.416	0.364
S+R Mode	0.461	0.398	0.159	0.493	0.389	0.472	0.108	0.418	0.362
MTurk MaxDiff	0.508	0.412	0.184	-	0.408	0.496	0.115	0.501	0.374
SampledBaseline	0.238	0.178	0.042	0.254	0.162	0.205	0.100	0.221	0.175
Random Baseline	0.239	0.186	0.045	0.249	0.269	0.200	0.110	0.269	0.196

Table 4.7: IAA between aggregated worker sense annotations and annotations of the GWS annotators (S+R denotes Select and Rate).

worker responses by measuring the IAA of the aggregated sense annotations with the annotations of the GWS annotators. For the scale-based ratings, Likert and Rate, we considered three arithmetic operations for aggregating the worker responses to produce the final applicability rating of each sense: mode, median, and mean. MaxDiff responses were aggregated using the counting procedure described in Section 4.3.5.

The worker IAA is compared with GWS annotators against sampled and random sense annotation baselines. In the sampled baseline, each sense is assigned a rating by sampling from the distribution of applicability rating used by the GWS annotators for that sense. The randomized baseline is created by assigning each sense an applicability uniformly sampled in $[1, 5]$. For each random baseline, we report the mean α across 50 sampled annotations of the data.

Results Table 4.7 reports the IAA between aggregated worker annotations and the annotations of the GWS annotators. Three findings emerge. First, aggregated annotations achieve moderate aggreement with the annotations of the GWS annotators. Furthermore, IAA with the GWS annotators increases proportional to worker agreement (cf. Table 4.6), indicating that when workers agree more, their combined annotations

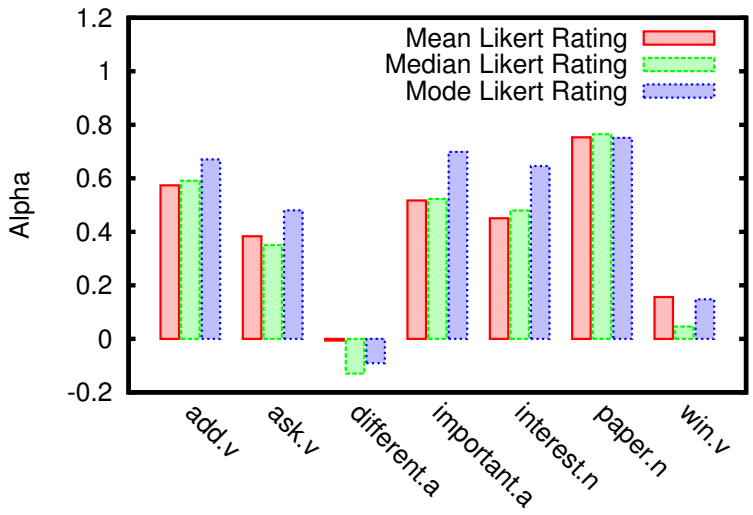


Figure 4.11: IAA between aggregated Likert and MaxDiff solutions, where the Likert ratings are aggregated using one of three methods.

better correspond to those performed in-person. Of the three methods, MaxDiff produced aggregate annotations that were most similar to those of the GWS annotators. Surprisingly, Select and Rate provided no advantage over Likert ratings, with half of the terms performing slightly better with each. All three methodologies outperform the baselines.

Second, for the scale-based annotation methodologies, the aggregation procedure had only moderate impact on the results. For Likert annotations, select the mode rating (i.e., most frequent) produced the highest average agreement. While for the Select and Rate annotations, the median yielded the highest - though the mode rating was similar in value. In both cases, the mean rating produced the worst performance, indicating that the workers' rating distributions often contained outlier ratings that distort the mean value away from the rating assign by other annotators.

Third, the GWS annotators had considerably lower agreement for some words (e.g., *win.v*), indicating these words were more difficult to annotate. In such cases, one of the GWS annotators could have a higher IAA individually with a method's aggregated

annotation, suggesting that that that annotator was producing more reliable annotations. However, no such trend is observed. Therefore, as a further comparison, we computer the IAA between the aggregated annotations of the Likert scale (with its mode rating) and the MaxDiff annotations, shown in Figure 4.11.

The agreement shown in Figure 4.11 provides three insights. First, the low IAA for *win.v* and *difficult.j* by the GWS annotators is not improved by using different annotation methodologies. This universally-low IAA suggests that the annotation guidelines themselves should be clarified, both for workers and annotators. Second, moderately-high IAA between Likert scale and MaxDiff aggregate annotations indicates that the different methodologies are capable of capturing the same information from workers. Furthermore, the mode rating of the Likert scale produces significantly higher agreement than the mean or median ratings, which further strengthens the case for using the mode rating when aggregating. Third, the IAA for four words is at least 0.67, which Krippendorff (1980) suggests as a minimum for drawing tentative conclusions, and approaches the level of agreement achieved for other sense-annotation experiments when using trained annotators (Kilgarriff and Rosenzweig, 2000; Snyder and Palmer, 2004; Mihalcea et al., 2004).

4.5 Experiment 3: Annotation Replicability

In the third experiment, we perform a replication study to assess the degree of reproducibility of the sense annotations by a different set of workers: Given an independent re-annotation of the same texts, what is the likely agreement between the two resulting aggregated solutions? Replicating the annotation procedure with unique sets of workers provides an indication of both the quality of the annotation and the clarity of the annotation procedure itself (Kilgarriff, 1999).

Replication was performed by sampling annotations from disjoint sets of workers and then aggregating each. Each question in the dataset was initially annotated by

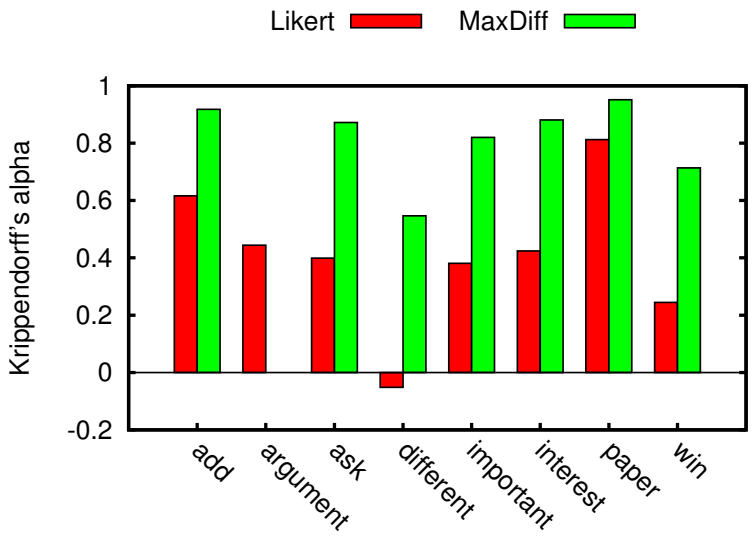


Figure 4.12: IAA between aggregated solutions produced from disjoint subsets of five workers.

Annotators	IAA	Subset IAA†
Erk et al. (2009) annotators	0.349	0.418
MaxDiff◇	0.373	0.419
Likert	0.241	0.295
Replicated Likert (Aggregated)	0.409	0.445
Replicated MaxDiff (Aggregated)◇	0.815	0.880
Aggregated Likert and Aggregated MaxDiff◇	0.472	0.649

† Excludes *win.v* and *difficult.a*, which had the lowest IAA

◇ Excludes agreement for *argument.n*, which was not annotated

Table 4.8: Worker and Annotator IAA rates for the GWS corpus

ten workers. Then, each item is assigned two sets of five worker annotations, sampling without replacement from the original set of annotations. Aggregated sense annotations are produced for each item from these two sets. We repeat this process 50 times and report the mean agreement measured using all items.

Results Figure 4.12 shows the mean α for each word when instance ratings are drawn from disjoint sets of five workers. Table 4.8 shows the α of all words for each data set. Two results are seen. First, MaxDiff annotations is highly replicable and five of the seven words achieve an $\alpha \geq 0.8$, which Krippendorff (2004) suggests is necessary to claim high-quality agreement. Indeed, when the lowest-performing lemmas of *difficult.a* and *win.v* are removed, α increases to 0.880 indicating that two independent sets of annotators are highly likely to produce the same sense annotations.

In contrast, aggregated solutions from Likert ratings show much less agreement, attaining an average $alpha$ of only 0.472 and achieving an $\alpha \geq 0.8$ for only one word, *paper.n*. However, this agreement surpasses that of the GWS annotators, suggesting that the methodology would be expected to perform the same for both workers and annotators. Using MaxDiff is more likely to generate a aggregated key with similar or identical distributions, with an average improvement in α over Likert aggregated of 0.3. However, an additional caveat for the IAA of Likert ratings is that five ratings may be too few to produce a reliable rating from the workers. Current approaches have typically used more than five workers because of the difficulty of the task and further work is needed to assess quality relative to the number of workers.

Second, MaxDiff replicability has agreement rates that are on par with those of previous expert-based annotation studies, shown in Table 4.9. While individual worker agreement is moderate, their aggregated sense annotations are high quality and replicable. In comparison with the studies in Table 4.9, the annotations from two independent sets of MaxDiff ratings have the second-highest IAA for WordNet-based studies.

4.6 Conclusion

Word sense annotation is an important but laborious task. Due to the need for high-quality annotations, experts and trained annotators have been responsible for creating all of the large-scale sense-annotated data sets in use, which creates a significant

Corpus	Sense Inventory	IAA	Measurement
SensEval-1 (Kilgarriff and Rosenzweig, 2000)	HECTOR	0.950	Replicability experiment (Kilgarriff, 1999)
OntoNotes (Hovy et al., 2006)	OntoNotes	$\geq 0.90^{\dagger}$	Pairwise agreement
SALSA (Burchardt et al., 2006)	FrameNet	0.86	Percentage agreement
SensEval-2 Lexical Sample (Kilgarriff, 2002)	WordNet 1.7	0.853, 0.710, 0.673‡	Adjudicated Agreement
SemCor (Fellbaum et al., 1998)	WordNet 1.6	0.786, 0.57*	Percentage agreement
SensEval-3 (Snyder and Palmer, 2004)	WordNet 1.7	0.725	Percentage agreement
MASC (Passonneau et al., 2012a)	WordNet 3.1	-0.02 to 0.88$^{\triangleleft}$	Krippendorff's α with MASI (Passonneau et al., 2006)
MASC, single phase reported in Passonneau et al. (2010)	WordNet 3.1	0.515	Krippendorff's α
GWS Corpus (Erk et al., 2009)	WordNet	0.349	Krippendorff's α

† not all words achieved this agreement

‡ Kilgarriff (2002) use a multi-stage agreement procedure where two annotators rate each item, and in the case of disagreement, a third annotator is added. If the third annotator agrees with either of the first two, the instance is marked as a case of agreement. However, the unadjudicated agreement for the dataset was 67.3 measured using pairwise agreement. A re-annotation by Palmer et al. (2004) produced a similar pair-wise agreement of 71.0.

* Ng et al. (1999) perform a re-annotation test of the same data using student annotators, finding this substantially lower agreement of 0.57, compared with the original 0.725

$^{\triangleleft}$ IAA ranges for 37 words; no corpus-wide IAA is provided.

Table 4.9: Inter-annotator agreement for sense-annotated corpora (top) and agreement for Turker-based annotations of the GWS corpus (bottom)

knowledge acquisition bottleneck. In this chapter, we test the potential for gathering sense annotations using crowdsourcing, where untrained online workers perform the annotation task. We propose two new annotation methodologies designed to simplify the task itself for untrained workers: (1) a two-stage methodology where one group of workers identifies potentially-applicable senses and then a second groups focuses only on those senses to provide detailed applicability ratings, and (2) a MaxDiff-based methodology where workers compare senses' applicabilities without the need for a rating scale. In three experiments, we compare our two new methododologies to the traditional single-sense methodology and the Likert scale methodology of Erk et al. (2009), providing two main contributions. First, we demonstrate that restricting work-

ers to using a single sense in annotation produces low IAA; but, by allowing workers to mark multiple senses, worker agreement rates rise substantially, approaching those seen with in-person annotators. Second, we demonstrate that the choice in annotation methodology significantly impacts the worker's ability to produce correct annotations, with MaxDiff offering superior performance. We demonstrate that MaxDiff produces highly-replicable annotations, with agreement rates in replicability tests on par with those seen between experts. Together, these results show the benefit of tailoring sense annotation to the level of experience appropriate for a crowdsourced worker rather than directly apply the common method used by experts.

CHAPTER 5

An Analysis of Ambiguity in Sense Annotations

Summary Ambiguity as to which sense best applies to a usage is a major source of difficulty for both annotators and WSD systems - especially when multiple senses appear to be valid interpretations. To better understand the phenomena of ambiguity between multiple senses, in this chapter we introduce a new WordNet sense-annotated corpus where all such cases with multiple interpretations are explicitly annotated, which was prepared as a part of SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013). We then use this corpus in our analysis of ambiguity, described in Jurgens (2014), to identify the causes by which annotators perceive multiple senses, the frequency of those causes, and how the ambiguity may be resolved. Our findings show that most ambiguity is due to insufficient contextual cues within a sentence but that a sizable minority of instances have contexts with distinct cues that each elicit different senses. Furthermore, we show that using a more coarse-grain sense inventory with fewer sense distinctions can reduce the need for multiple WordNet senses, but not eliminate it entirely.

5.1 Introduction

A word may take on a variety of meanings depending on the context. Word sense inventories formalize these meanings into discrete units, known as senses. Annotators later consult these senses to specify which meaning is present in a given context. In some cases, a word's senses may be related. For example, in the commonly used WordNet (Fellbaum, 1998) and OntoNotes (Hovy et al., 2006) sense inventories, *bank* may refer

to the financial institution, the building that the institution occupies, or an object used to store money (e.g., a piggy bank). These related senses can cause difficulty for annotators when determining the particular sense of a word's usage. Indeed, OntoNotes even includes a special sense of *bank* to indicate when the usage is ambiguous between the institution and building senses.

While most annotation efforts restrict usages to having a single sense even when ambiguous, another possibility is to label such usages with multiple senses in order to explicitly model the multiple interpretations. Previous annotation studies have shown that annotators will use multiple senses if allowed (Véronis, 1998; Murray and Green, 2004; Erk et al., 2009; Passonneau et al., 2012b; Jurgens, 2013). However, little work has assessed the underlying factors causing annotators to perceive multiple senses and what the resulting interpretation of the usage is with its multiple labels. Furthermore, no study has proposed guidelines for when annotators should use multiple senses, despite expressing the need for them (Krishnamurthy and Nicholls, 2000; Langone et al., 2004).

This chapter focuses on three open questions: (1) what is the frequency with which a lemma may have valid, differing semantic interpretations, (2) what contextual factors contribute to the perception of multiple senses and what are their relative frequencies, and (3) what is the relationship between the annotation's multiple senses and how is the usage interpreted as a result. To answer these questions we constructed of a new sense-annotated corpus used in SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013) where word usages are explicitly annotated with the applicability of each of its WordNet sense in the given context.

Our work offers four main contributions. First, we produce a new multiple-sense annotated corpus and describe its construction procedure. Second, we demonstrate that contextual underspecification is responsible for instances with multiple sense annotations in nearly two-thirds of the cases, while the remaining cases are due to syllepsis,

where multiple contextual features each select different senses.[1] However, verbs are more likely to appear in sylleptic contexts. Third, we show that the fine granularity of sense distinctions in WordNet is not completely responsible for the perception of multiple senses: While nearly all verb instances with multiple senses would be annotated with a single OntoNotes sense, for nouns, conflicting interpretations or cases of regular polysemy would still require the instance to have multiple OntoNotes senses in approximately 23% of cases. Fourth, we propose a two-way classification scheme, which we argue would serve sense annotators as a guideline when considering multiple senses for a usage.

5.2 Related Work

Work on ambiguity in word sense annotation has often focused on techniques to reduce ambiguity in the sense inventory in order to improve annotator agreement. Most work has addressed the aspect of sense granularity, with many proposals for how to reduce ambiguity by adapting a sense inventory to make its senses hierarchical (Edmonds and Cotton, 2001), underspecified (Buitelaar, 2000), or more coarse-grained (Palmer et al., 2004; Palmer et al., 2007). Other work has proposed creating new sense inventories around annotators' perceived distinctions in meaning (Rumshisky and Batiukova, 2008; Biemann, 2012).

Several works have investigated allowing annotators to use multiple senses. Véronis (1998) analyzed 600 hundred contexts each for 60 French words using a common French dictionary to define their senses. The author notes that the average rate of multi-sense annotation was low, with an average of 1.02 senses per instance, but was higher

[1]Here, syllepsis refers to the linguistic phenomena where a word is used in relation to two or more parts of the sentence and must be understood differently in relation to each. For example, in the adage, "Chocolate, coffee, and men: the *richer* the better," the lemma *rich* is interpreted differently for each of the three words it modifies. Syllepsis may also elicit significantly different interpretations of the same word, e.g., "you are free to execute your laws and your citizens as you see fit," (from *Star Trek: The Next Generation*).

for some words (e.g., an average of 1.311 senses per instance for the verb *comprendre*). Furthermore, the author notes that disagreements were not resolved by merging similar senses, with most disagreement occurring between much coarser senses. To fully capture all perceived senses, Erk et al. (2009) had three annotators label 50 contexts each for eight words, rating all senses of the word using a scale from one (inapplicable) to five (completely applicable). Their annotators readily used multiple senses per instance, with 78.8% of usages having multiple senses rated at least a three on their scale. Similar to the study of Véronis (1998), Erk et al. (2009) found that the presence of multiple sense could not be reduce to a single sense by grouping highly-correlated senses, with over 40% of contexts having two senses that were not correlated. Last, Alonso et al. (2013) asked crowdsourced workers and volunteers to annotate between literal and metonymic senses, allowing them to select a third option of "both" in cases of underspecification. Annotators for Danish, English, and Spanish all used the underspecified option, demonstrating that, in conjunction with the French study of Véronis (1998), the perception of multiple senses is common across languages and sense inventories.

The most similar to the proposed study is that of Passonneau et al. (2009), who analyzed factors contributing to annotator disagreement, finding that agreement was often mediated by the lemma itself and its usages. Looking across multiple lemmas, they found that the factors of sense similarity, contextual specificity, and sense concreteness most affected agreement. The present work offers a complementary analysis for understanding cases where multiple senses meaningfully apply, which potentially may have been treated as cases of annotator disagreement when usages were required to be labeled with a single sense.

5.3 Corpus Description

Nearly all sense-annotated corpora require annotators to use a single sense when annotating an instance. This single-sense annotation masks any potential alternate in-

	Number of Senses		
	Minimum	Mean	Maximum
Nouns	4	9.05	14
Verbs	4	8.60	22
Adjectives	4	7.60	10

Table 5.1: Polysemy statistics of the target words in SemEval-2013 Task 13

terpretations of a usage or ambiguity that an annotator might have. Therefore, as a part of Samovar-2013 Task 13 (Jurgens and Klapaftis, 2013), we prepared a corpus of multiple instances of fifty target words, allowing for annotators to label each instance with multiple senses, weighted by their applicability. Following, we describe the target words around which the corpus is built (Sec. 5.3.1) and then the annotation procedure (Sec. 5.3.2).

5.3.1 Corpus and Target Words

To focus the corpus on lemmas that may have multiple interpretations, potential target lemmas were drawn from two sources: (1) lemmas used in prior SemEval WSD tasks that had related senses and that yielded lower performance for WSD systems was lower than other lemmas in the task (Edmonds and Cotton, 2001; Snyder and Palmer, 2004; Mihalcea et al., 2004; Navigli et al., 2007), and (2) lemmas that were annotated in the MASC corpus (Passonneau et al., 2012a) and caused difficulties for that project's annotators.[2] The final list of target lemmas was then selected from these potential lemmas in order to balance the distribution of lemma polysemy, summarized in Table 5.1. Furthermore, while prior SemEval sense-based tasks have used only nouns and verbs, we include adjectives in our target lemmas. Table 5.2 lists the fifty selected target lemmas.

Contexts for the target lemmas were then drawn from the Open American National

[2]Rebecca Passoneau, *personal communication.*

Nouns		Verbs		Adjectives
board	paper	add	meet	common
book	part	appear	read	dark
color	people	ask	serve	familiar
control	power	become	strike	late
date	sight	book	suggest	new
family	sound	dismiss	trace	poor
force	state	find	transfer	serious
image	trace	help	wait	severe
life	way	live	win	strong
number	window	lose	write	warm

Table 5.2: Target words used in SemEval-2013 Task 13

Corpus (OANC) (Ide and Suderman, 2004). Contexts were selected from multiple genres, which enables measuring multiple-sense annotation rates in different contexts, summarized in Table 5.3. An automatic process was used to sample an initial list of usages for each of the target lemma, which were then manually inspected to ensure that (1) the usage had not been automatically tagged with the wrong part of speech, (2) the usage could be interpreted with at least one WordNet 3.1 sense, and (3) the usage was not part of an idiomatic usage (e.g., "kick" in "kick the bucket"), named entity ("family" in "ABC Family") or similar types of non-compositional multiword expressions. Ultimately, 4664 contexts were used as test data, with a minimum of 22 and a maximum of 100 contexts per word. Contexts were restricted to the sentence containing the usage of the target lemma. This size matches common real-word settings such as microtext domains like Twitter, where limited context for disambiguation is available (Gella et al., 2014).

5.3.2 Sense Annotation

Usages of the target lemmas were annotated by the two organizers of Task 13 (Jurgens and Klapaftis, 2013). The first author (myself) annotated all instances and the

	Spoken		Written						
Genre	Face-to-face	Telephone	Fiction	Journal	Letters	Non-fiction	Technical	Travel Guides	*All*
Instances	52	699	127	240 3	103	477	611	192	4664
Tokens	1742	30,700	3438	69,479	2238	11,780	17,337	4490	141,204
Word Types	577	3559	1498	14,084	854	3771	4771	2035	19707
Mean Tokens per Context	39.4	45.8	30.9	33.7	24.3	27.5	31.7	26.4	34.1
Mean senses/inst.	1.17	1.07	1.12	1.11	1.30	1.08	1.09	1.09	1.10

Table 5.3: Statistics for the annotated corpus used in Jurgens and Klapaftis (2013), according by genre

second author annotated a 10% sample of each lemma's instances in order to estimate Inter-Annotator Agreement (IAA). Items were rated using the Likert scale annotation procedure described in Chapter 4 Section 4.3.3), following the guidelines of Erk et al. (2009). During this author's annotation effort, all instances of a target lemma were annotated sequentially and then the same instances were rechecked to correct errors and ensure a consistent application of the sense applicability ratings. IAA was calculated using Krippendorff's α for Likert scales as in Chapter 4.

The total IAA for the dataset was 0.504, and on individual words, ranged from 0.903 for *number.n* to 0.00 for *win.v*. While this IAA is less than the 0.8 recommended by Krippendorff (2004), it is consistent with the IAA distribution for the sense annotators working on other parts of the OANC corpus; Passonneau et al. (2012a) reports an α of 0.88 to -0.02 for trained sense annotators in the MASC subset of the OANC. Furthermore, a re-analysis of the annotations showed that disagreements were often due to incorrect interpretations from the second author; thus, the annotation quality is likely to be higher than indicated from IAA alone.

The bottom row of Table 5.3 summarizes the sense annotation statistics for the data set. Rates of using multiple-sense annotations were similar across the genres of the OANC. The rate of multiple sense annotation is weakly negatively correlated with average context size across genre, with Pearson $r = -0.374$, though the number of data points is too small to use in drawing statistically significant conclusions. Due to the variety of genres from which contexts were drawn, all lemmas were observed

	Parallel	Admissible	Conflicting	*Total*
Underconstrained	47	158	92	297
Sylleptic	122	26	15	163
Total	169	184	107	460

Table 5.4: Distribution of assignments to the Context and Sense Assignment classifications

with at least two distinct senses, with many lemmas with at least half of their senses observed.

5.4 Classification Schema

The process of determining the appropriate sense for a usage is a function of both the context and the options available in the sense inventory. Therefore, we analyze each usage with a two-way classification schema that incorporates both factors. The first classification axis assesses the type of semantic constraints put upon the usage by the context. This axis captures cases of ambiguity from too few contextual cues and cases of syllepsis where multiple dependent clauses or modifiers refer to different meanings of the same usage. The second classification axis assesses the types of senses that are elicited by the usage and how those senses together may be interpreted within in the context. For example, this axis captures cases where the context evokes highly-similar senses that do not change the overall meaning of the context. We refer to these classification types as Context and Sense Assignment, respectively. Annotation guidelines were developed for both types iteratively through a close analysis of 50 randomly-selected instances and then applied to annotate all instances. After a first pass of annotation, each instance was then re-annotated to correct errors and ensure the guidelines were clear. Following, we formalize the classifications and present examples from the corpus. For clarity, the sense descriptions with each example also include the

corresponding WordNet sense keys used in the dataset.

5.4.1 Context Classification

Contexts were divided into two classes: (1) those containing too few semantic cues to constrain the usage to a single interpretation and (2) those exhibiting semantic syllepsis where the dependent clauses or modifiers of a usage require it to be interpreted with different meanings. We refer to these classes as Underconstrained and Sylleptic, respectively.

Underconstrained contexts may occur due to contexts having too little content as a whole and due to contexts omitting specific information needed to distinguish between related senses. We illustrate these through two examples. In the following context for *warm*:

> Rooms are classically decorated and *warm*.

the cues enable interpretations of a comfortable level of heat (warm%3:00:01::) and being colored in such a way to evoke warmth (warm%3:00:03::), despite the large difference in the senses' meanings. In contrast, consider the context for *find*.

> The Random House lexicographer Jesse Sheidlower has *found* a reference to it in a passage from Varieties of Religious Experience, in which William James quotes words of Voltaire, for which he gives the date 1773: "All comes out at the end of the day, and all comes out still more even when all the days are over."

The interpretation of *find* is dependent upon the mental state of the actor and state of the passage in question. The context enables readings where the individual unintentionally encountered the reference (find%2:40:02::), was actively seeking the reference out (find%2:39:02::), or had rediscovered a lost reference (find%2:40:00::). For both

example contexts, additional cues can restrict the interpretations to a single sense, e.g., adding "in winter" as a prepositional phrase modifying *warm* in the first example.

Sylleptic contexts frequently occur from two constructions. The first construction occurs when a word is associated with multiple senses, each describing different aspects of the same concept. For example, *chicken* may refer to both the animal and the food. In many cases, classes of words exhibit the same type of semantic distinctions (e.g., Animal/Food), which are known as cases of regular polysemy (Apresjan, 1974; Pustejovsky, 1995; Ravin and Leacock, 2000). These Sylleptic constructions evoke the same concept through having clauses or modifiers refer to its different aspects. We illustrate this construction with an example for *book*.

> The fat *book* here surveys the hundreds of books and articles already written about Rockefeller

The usage evokes both the physical object (book%1:06:00::) by means of the adjective fat and the book's role as a literary work (book%1:10:00::) by the discussion of its content. This object-role sense distinction is seen in other lemmas with the same type of regular polysemy, such as *magazine* or *paper*.

The second common type of Sylleptic construction comes from metaphoric usages that evoke both literal and figurative meanings of the word. We illustrate this with an example for *dark*.

> We all are relieved to lay aside our fight-or-flight reflexes and to commemorate our births from out of the *dark* centers of the women, to feel the complexity of our love and frustration with each other, to stretch our cognition to encompass the thoughts of every entity we know.

By their nature, the centers of women are devoid of light (dark%3:00:01::), but the usage also allows a reading with a secretive connotation (dark%3:00:00:concealed:00) to the role of the darkness.

5.4.2 Sense Assignment Classification

The fine-grained sense inventory of WordNet permits some sense interpretations to be concurrently true (e.g., describing a lemma's purpose and physical properties). However, some contexts elicit conflicting interpretations. Therefore, the second classification axis describes the consistency of the usage's interpretation according to three classifications: (1) the senses may be concurrently true (2) the senses are sufficiently related to evoke a basic interpretation that could be further refined to one sense, given more context, and (3) the senses describe conflicting interpretations of the usage in context. We refer to these classifications as Parallel, Admissible, and Conflicting, respectively, and following illustrate each with examples.

Parallel interpretations may be caused both by Underconstrained and Syllpetic contexts. Consider the usage of *severe* below.

> The shock was *severe* enough to strike her dumb, and she was committed to a London hospital.

The context allows interpretations of *severe* with respect to two aspects of the sentence, describing the shock as intensely bad or unpleasant (severe%3:00:00:intense:00) or the shock's effects on the recipient as grievous or causing harm (severe%3:00:00:critical:03). Both interpretations may be valid within the interpretation of the entire context.

Admissible interpretations reflect cases where a usage's senses express fine-grained distinctions or closely-related concepts. For example, consider the following use of *family*.

> He added that his wasn't a dysfunctional *family*.

Here, the scope of *family* could refer to both the immediate family (family%1:14:00::) or to a larger unit of blood relatives (family%1:14:01::). However, the context can still be interpreted with approximately the same meaning without resolving the ambiguity, i.e., *family* referring to people that are related in some way to the sentence's subject.

		Nouns		Verbs		Adjs.
Context	Sense Assignment	%	JCN	%	JCN	%
Underconstrained	Admissible	43.9	0.151	22.5	0.091	35.9
Underconstrained	Conflicting	24.4	0.064	16.6	0.076	18.6
Underconstrained	Parallel	4.9	0.131	5.2	0.101	21.4
Sylleptic	Admissible	1.8	0.073	11.9	0.100	3.4
Sylleptic	Conflicting	3.0	0.062	4.6	0.112	2.0
Sylleptic	Parallel	22.0	0.098	39.1	0.065	18.6

Table 5.5: Distribution of assignments to the two-way classifications with the percentages of total instances per part of speech and the average JCN similarity of senses assigned to each instance having that classification.

In Conflicting cases, the ambiguity in a usage's sense elicits distinct incompatible interpretations. Consider the meaning of *image* below.

> Thomason and the White House aren't talking, so I consulted my own *image* expert, Jackson Bain.

Without aid of further context, it is unclear whether the individual's expertise pertains to visual representations such as photographs (image%1:06:00::) or to managing a public persona (image%1:07:00::).

5.5 Results and Discussion

The proposed annotation scheme was applied to all 542 instances with multiple sense assignments in the dataset of Jurgens and Klapaftis (2013). During annotation, our analysis suggested that 82 instances (15%) containing a sense rated as having low applicability should be excluded from the analysis, as these low-applicability sense would not be considered valid interpretations of the usage according to our guidelines. The

remaining 460 annotated instances were used in our analysis. Instances were approximately equally distributed across part of speech classes, with 164, 151, and 145 instances for noun, verb, and adjective lemmas, respectively. Table 5.4 shows the distribution of instances across the two-way classification scheme.

Four analyses were performed to test (1) differences in ambiguity according to part of speech, (2) the relationship between an instance's senses' similarity and its classification type, (3) the effect of sense granularity on sense ambiguity, and (4) the presence of lemma-specific preferences towards certain types of ambiguity. Following, we discuss general observations of the instances' classifications and then describe the results of each analysis. We conclude with a discussion of the how current sense annotation guidelines might be improved.

5.5.1 General Observations

Three general trends appear across all instances. First, Underspecified contexts are nearly twice as common as Sylleptic constructions. Because contexts are single sentences, the high frequency of Underspecified contexts raises the possibility that the majority of ambiguous cases could potentially be resolved to a single sense using additional context outside the sentence. Second, Underspecified and Sylleptic contexts can cause all types of sense classifications, though each context type has a clear preference. Third, only 23% of the instances have Conflicting semantic interpretations, suggesting that in the majority of multiple-sense cases, a correct interpretation of the entire context is not dependent upon refining the annotation to a single sense.

5.5.2 Part of Speech

The first analysis measures the differences in the classifications according to part of speech. Table 5.5 shows the percentage of instances assigned to each. Clear distinctions between part of speech classes emerge. Both nouns and adjectives are much more likely

to be in Underconstrained contexts (73.2% and 75.9%, respectively), while verbs are slightly more likely to be in Sylleptic contexts (55.6%).

Within the observed word types, we observed a trend in adjectives and nouns where types had highly-related senses that were difficult to distinguish between with the available context. For example, in the following instance of *new*,

> By reengineering business processes in conjunction with implementing *new* technology, Owens Corning increased its ability to meet customer needs.

the context makes it unclear whether the new technology has recently been invented (new%3:00:00::) or is simply unlike the previous technology (new%3:00:00:other:00). Similarly, in the early example of *family*, distinguishing between the immediate and extended family senses is difficult in shorter contexts, though the former is a more probable interpretation. However, the number of word types in the dataset is too small to make strong generalizations about the behaviors of each part of speech class.

The frequency with which instances take on multiple senses also varies by part of speech, with 10.1% of noun instances, 8.1% of verb instance, and 15.1% of adjective instances having multiple instances. The increase in frequency for adjectives over the rates for nouns and verbs is statistically significant at $p<0.01$. We hypothesize that adjectives show an increased frequency because they may take on many different shades meanings, depending on the noun (Pustejovsky, 1995) and sparse contexts increase the difficulty in selecting only one of these related meanings.

Because the lemmas used in the dataset of Jurgens and Klapaftis (2013) were intentionally selected based on having exhibited sense ambiguity in previous annotation studies, the frequencies with which lemmas have multiple senses are likely to be higher than those for a larger sample of lemmas. However, the frequencies may still provide soft upper bounds for those expected in larger corpora and could be useful for identifying annotators who are over-zealous in using multiple senses.

5.5.3 Sense Similarity

Fine-grained sense distinctions are often reflected in the high degree of similarity between two senses of a word. We hypothesize that sense similarity may reveal whether fine-grain distinctions contribute to certain types of ambiguity as expressed in the classifications. Therefore, in the second analysis, for each noun and verb instance, we calculate the similarity of its senses using the Jiang and Conrath (JCN) similarity measure (Jiang and Conrath, 1997), which was shown to most-closely approximate human similarity judgments (Budanitsky and Hirst, 2006). The JCN columns in Table 5.5 show the average similarity for nouns and verb instances assigned to each classification type.

Among the nouns, two trends emerge. First, the selected senses are more similar when a context is Underconstrained than when Sylleptic, which highlights the differences in the underlying mechanisms. In Underconstrained contexts, the lack of semantic cues makes distinguishing between similar senses difficult, resulting in multiple senses in the annotation, while Syllpetic contexts are more likely to evoke senses related to a single concept which are not necessarily themselves similar. Second, instances with a Conflicting annotation classification have sense with lower similarity than those of the other two classifications; however, we note that the number of Sylleptic nouns instances is too few to draw statistically significant conclusions with respect to sense similarity.

The sense similarity of verb instances did not show the same trends as noun instances. Though minor difference in verb sense similarities are present, differences between classifications are not significant at $p<0.05$.

5.5.4 Sense Granularity

In our third analysis, we consider whether multiple interpretations could be due to the granularity of the sense inventory. To test this hypothesis, each instance's WordNet annotation was evaluated to see whether its multiple senses were all subsumed by a single,

Context	Sense Assign.	Nouns	Verbs
Underconstrained	Admissible	90.3	97.1
Underconstrained	Conflicting	57.5	100.0
Underconstrained	Parallel	87.5	100.0
Sylleptic	Admissible	100.0	100.0
Sylleptic	Conflicting	60.0	100.0
Sylleptic	Parallel	69.4	100.0

Table 5.6: Percentage of instances with multiple WordNet senses that would receive a single OntoNotes sense label

more coarse-grained OntoNotes sense (Hovy et al., 2006), using the publicly-available mapping between WordNet and OntoNotes senses for nouns and verbs. As OntoNotes senses are more coarse-grained, frequently many WordNet senses are mapped into a single OntoNotes sense Table 5.6 lists the percentage of instances for each classification and part of speech whose multiple senses would be represented by a single OntoNotes sense.

The percentages of instances that would be annotated with a single OntoNotes sense, shown in Table 5.6, reveal two clear differences between parts of speech and the classifications. First, nearly all verbs (99.3%) would be annotated with a single OntoNotes sense. WordNet often contains fine-grain sense distinctions of a single action for a verb, reflecting aspects such as the circumstances, effect, and implications related to the action. For example, the verb *transfer* contains different senses for the action of moving an object depending on whether the movement implies a change of ownership. In the following instance, it not clear whether the authorities provided copies of their documents (transfer%2:40:01::) or provided the originals such that they no longer have ownership (transfer%2:40:00::):

"The US Administration stated on January 12 that all documents on 'the

Iranian case' have been *transferred* to the competent Russian agencies through diplomatic channels," Segodnya said.

Indeed, OntoNotes merges six WordNet senses of *transfer* relating to this action into one sense. However, coarsening the senses loses information. In the above example, an OntoNotes annotation would not longer convey whether documents were retained by the US Administration, which could potentially affect downstream applications using senses, such as Textual Entailment.

In the second difference between parts of speech, sense coarsening reveals that Admissible contexts were nearly always represented with a single OntoNotes sense for nouns and verbs, irrespective of the context classification, but in contrast, a large minority (23.2%) of Conflicting and Parallel noun instances would still require multiple OntoNotes senses. The latter minority is due to cases where the instance requires significantly different interpretations (e.g., the earlier example instance of *image*) or where OntoNotes does not merge cases of regular polysemy into a single sense.

5.5.5 Distribution of Annotations

Given the differences in classification distribution per part of speech, in the fourth analysis, we assess whether the classification types were also unevenly distributed across the lemmas themselves; i.e., whether all instances of a lemmas occurred with the same classification type. Figure 5.1 shows the distribution of the six classification combinations across all 49 lemmas having multiple sense annotations. Lemmas had on average 3.02 different classifications across their instances. Furthermore, the number of classes seen was moderately correlated with the total number of instances per lemma, Pearson's r=0.468, which is statistically significant at $p<0.01$. Together, these suggests that most lemmas have sense distinctions that may be ambiguous in certain circumstances and that given enough instances of the lemma, its usages expressing multiple interpretations will be seen with a variety of classification types.

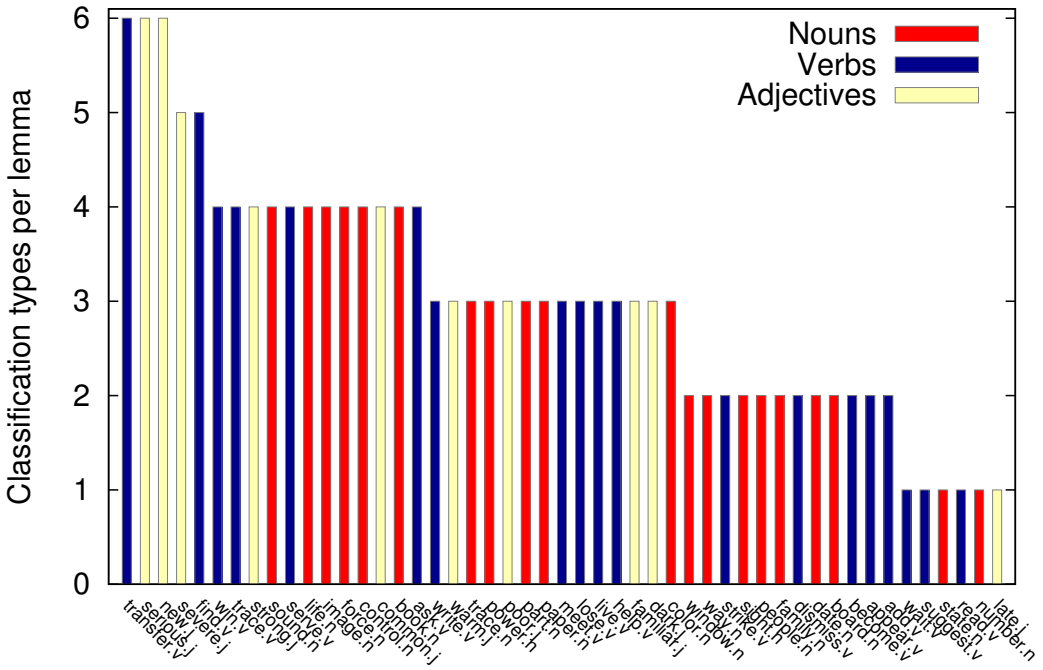


Figure 5.1: A comparison of the number of unique Context-Sense Assignment classifications seen per lemma

5.5.6 Sense Annotation Recommendations

The two-way classification was designed with an additional motivation of providing guidance to sense annotators when deciding whether an instance should have multiple senses. While prior annotator efforts have allowed annotators to use multiple senses, no guidelines have been proposed for when annotators should use them, despite an expressed need (Langone et al., 2004). For example, Krishnamurthy and Nicholls (2000) note ambiguous sylleptic examples for *onion*, stating that during the annotation effort for SensEval-1, annotators chose only a single sense of these using their own *ad hoc* heuristics.

Our first recommendation is that sense annotators should be encouraged to assign multiple senses but do so only in cases where they are able to additionally classify

the usage according to our proposed classification schema. This recommendation is motivated by two factors. First, the process of classifying the instance encourages annotators to scrutinize the context in order to justify why multiple senses would be present. This discourages the introduction of multi-sense annotations due to an annotator accidentally overlooking sense-specific cues. However, the classification process is sufficiently lightweight and related to the main task so as not to overburden annotators. Second, the classifications can be important for later applications needing to interpret the multiple senses. Similarly, the classifications can be of use for the evaluation of Word Sense Disambiguation (WSD) systems. In cases of an Underspecified context, a WSD system's output could be considered correct if it contains any of the multiple senses; however, in Sylleptic contexts, the WSD system would need to recognize all the intended meanings.

Our second recommendation is that in cases of Underconstrained contexts, annotators should be able to mark which sense is more likely. We motivate this with an example for *strike*.

> Traditionally eight bells are *struck*.

The likely meaning of the context is that the bells were hit to make a sound (strike%2:35:00::); however, in certain domains, the eight bells could have been forged (strike%2:36:02::), perhaps in commemoration. In the absence of available cues, annotators can use their background knowledge to indicate which sense is more probable. We note that this weighting differs from the sense applicability ratings used by Erk et al. (2009) for labeling multiple senses per word; in our setting, both senses would already be applicable, with the weight specifying only their likelihoods of being the correct interpretation. The proposed weights could be used by downstream applications to select a single interpretation and by WSD evaluations to favor a system that reports the more probable sense, much like prior suggestions for evaluating based on sense similarity (Resnik and Yarowsky, 2000) or applicability (Jurgens, 2012).

5.6 Conclusion

Word sense annotation is a challenging task where annotators may find multiple valid interpretations of a usage, leading to multiple senses in the same annotation. The present study has analyzed 460 instances annotated with multiple senses from the dataset of Jurgens and Klapaftis (2013). We then classified each according to a proposed two-way classification to quantify (1) the contextual features contributing to the multiple interpretations and (2) the impact of multiple senses on the interpretation of the usage in context. Our work offers four main contributions. First, we developed a large sense-annotated corpus comprising 4,664 instances for 50 target words total. Instances are annotated with all applicable senses, with 11% having multiple senses. Second, using the multiple-sense instances, we demonstrate that in two-thirds of those cases, the lack of contextual cues causes the perception of multiple senses, whereas syllepsis accounts for the remaining. However, verbs are still more likely have multiple senses from sylleptic constructions, indicating that part of speech must be taken into account. Third, analyzed the effects of fine-grained sense distinctions in WordNet, showing that for all verbs, each verb instance with multiple WordNet senses would have only a single OntoNotes sense. However, the same was not true for nouns, with approximately 23% still requiring multiple OntoNotes senses. Last, we introduced a two-way classification schema for labeling why an instance has multiple senses and how those senses may be interpreted together. We propose that this schema would serve future sense annotators as a guideline when considering whether to apply multiple senses for a usage, thereby improving annotation quality.

CHAPTER 6

Model Evaluation

Summary This chapter presents two experiments in quantitatively evaluating the performance of Word Sense Induction (WSI) and Word Sense Disambiguation (WSD) systems, including the two WSI approaches described earlier in Chapter 2. The first experiment analyzes performance on SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013), which includes (1) an evaluation measuring the ability of a WSI system to match lexicographer performance at grouping usages according to their WordNet senses, and (2) a WSD evaluation where instances may be labeled with multiple senses. In three analyses on the data set, we demonstrate that WSID systems offer superior performance to both unsupervised and supervised WSD systems. Furthermore, we show that our SWCD achieves state-of-the-art WSI performance at grouping instances by senses. In the second experiment, we evaluate the ensemble WSID approach of Chapter 3 on multiple, diverse sets of WSI models and using both WordNet and OntoNotes sense inventories. Using prior SemEval WSI tasks, we demonstrate that our ensemble WSID configuration consistently achieves a statistically significant performance improvement over the best system and baselines for each task.

6.1 Introduction

Sense induction methods serve two purposes within NLP. First, a WSI method may act as a computational lexicographer and group instances according to their senses. This grouping process automatically annotates usages with induced senses and provides a

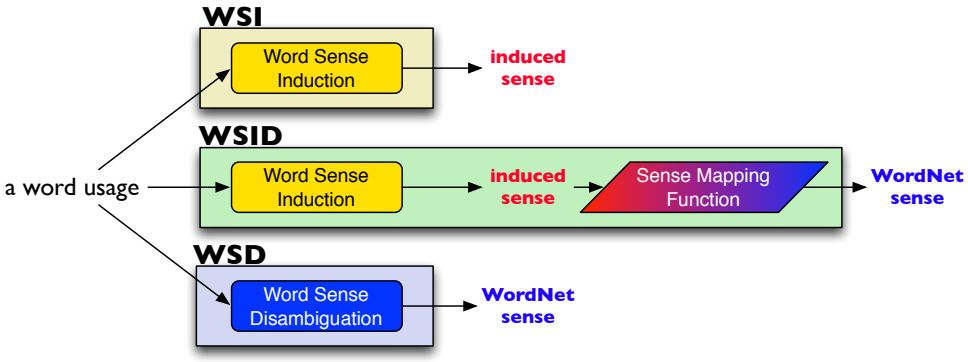


Figure 6.1: A comparison of Word Sense Induction and Disambiguation systems, showing the key components and outputs of each.

sense-annotated dataset for subsequent NLP applications that require senses or for lexicographers wanting to analyze particular words. Second, the induced sense annotations from WSI methods can be used as features for constructing WSID systems. These WSID systems provides a solution to the knowledge acquisition bottleneck for WSD by combining small amounts of manually-annotated data with the induced senses to produce a high-performance semi-supervised WSID system without the need for creating a large sense-annotated corpus.

Accordingly, a WSI system may be evaluated according to each of its uses: (1) as a system for grouping word usages according to their meaning, and (2) within a WSID system for disambiguating usage. To highlight the differences between the uses of WSI, Figure 6.1 illustrates the core components of WSI, WSID, and WSD systems and their respective outputs.[1] WSI systems produce induced sense annotations; therefore, WSI evaluations test the quality of these automatically-learned senses by measuring the degree of similarity between the sense distinctions found by the WSI system and those found by lexicographers (i.e., the degree to which a WSI system and lexicographers agree that certain usages have the same meaning). WSID and WSD systems produce sense annotations from a reference sense inventory (shown as WordNet senses in Figure

[1]For the purposes of the example, WSID and WSD systems are shown producing WordNet senses; however, senses from any other reference sense inventory, such as OntoNotes, could be used as well.

6.1); therefore, both types of systems may be evaluated according to whether their annotations exactly match the gold standard.

Previously, in Chapter 2, we proposed two WSI methods and then demonstrated qualitatively that they identify meaningful sense distinctions. Then in Chapter 3, we demonstrated using pseudoword data (1) that our SWCD method achieves the highest performance of graph-based WSI methods and (2) that our ensemble WSID approach provides a large performance improvement over WSD state of the art. In this chapter, we perform two quantitative evaluations of the abilities of these model on real-world data and compare their performance against current state-of-the-art WSI and WSD systems.

In the first evaluation, we test our WSI methods on SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013), which measures the systems' ability in both WSI and WSID settings. This evaluation compares our methods against the other WSI methods participating in Task 13 and also additional state-of-the-art supervised and unsupervised WSD methods. In three analyses on the Task 13 data, we demonstrate that (1) our SWCD method achieves state-of-the-art WSI performance at matching lexicographers' gold-standard groupings of instances according to sense, and (2) WSID methods, including our SWCD and AWCD methods, offer substantial performance increases over existing supervised and unsupervised WSD systems. In the second evaluation, we measure the benefits of our ensemble approach on three heterogeneous groups of WSI systems and two different sense inventories by using three SemEval WSI tasks (Agirre and Soroa, 2007; Manandhar et al., 2010; Jurgens and Klapaftis, 2013). We demonstrate that in all three tasks, combining WSI systems using our ensemble approach achieves a statistically significant performance improvement over current state of the art.

In what follows, we first introduce the evaluation measures used in Task 13 for our first experiment (Sec. 6.2). Then, after describing the data and systems used in our first experiment, we perform three forms of analysis: (1) analyzing performance on the full Task 13 data set, which includes instances labeled with multiple senses (Sec. 6.3.5),

(2) analyzing performance on the subset of instances (89%) labeled with only a single-sense using a modified test setup (Sec. 6.3.6), and (3) analyzing the sense mapping function used in Task 13 compared with a linear kernel SVM (Sec. 6.3.7), which we proposed in Chapter 3. Last, in our second experiment (Sec. 6.4), we demonstrate that our ensemble WSID method provides a consistent performance improvement in tests using multiple types of WSI systems and sense inventories.

6.2 Multi-Sense Evaluation Measures for WSI and WSD

As shown in Chapter 5, a usage of a word may have multiple interpretations, which can be modeled by labeling the usage with more than one sense. However, prior evaluations of WSI and WSD systems have use data that is annotated with only a single sense per instance (Agirre and Soroa, 2007; Manandhar et al., 2010) with corresponding evaluation measures that operate with only one sense per instance. Therefore, we propose extensions to current methods for evaluating the quality of both WSI and WSD systems when instances may be labeled with multiple senses, described next.

6.2.1 WSI Evaluation Measures

Word Sense Induction can be viewed as a clustering task: Given a set of instances, group them into clusters such that instances having the same sense are in the same cluster. To measure the quality of a WSI system, a comparison is made between the clusters formed for a particular set of instances by the induced senses and the clusters formed by the gold standard annotation of that data set. The degree of similarity between the two groups of clusters, referred to as *clusterings*,[2] reflects the ability of the WSI system identify the same sense distinctions as in the gold standard.

Figure 6.2 illustrates this type of clustering comparison with a set of nine instances.

[2]For stylistic reasons, we will occasionally refer to a clustering as a "grouping" or "group of clusters," with no loss in meaning.

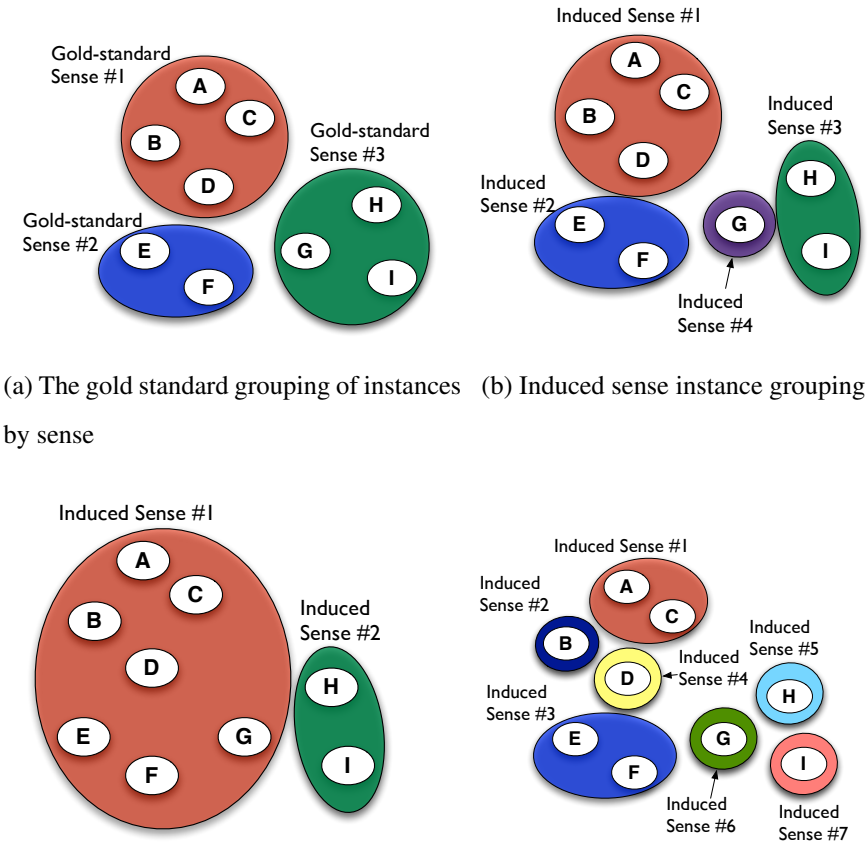
Figure 6.2a shows the nine instances clustered into one of three senses according to a gold standard (i.e., all instances in a cluster are annotated with the cluster's corresponding sense). Figures 6.2b–6.2d show three alternate groupings of the same instances according to how they were disambiguated with induced senses. Of the three induced sense groupings, grouping #1 appears closest to matching the gold standard, differing only for instance G. In contrast, grouping #2 merges Gold-standard Senses 1 and 2 as well as an instance assigned to Gold Standard Sense 3, while induced sense grouping #3 assigns many instances to their own senses.

The example in Figure 6.2 also highlights the difference between WSI evaluation and WSID evaluation, which was used in Chapter 3. WSI evaluations seek to test the degree to which the sense distinctions of the induced senses match those of the gold standard. In contrast, WSID evaluations test the degree to which induced sense annotations may be used to predict the gold-standard annotations. For example, although Gold Standard Senses 1 and 3 are divided into fine-grain clusters in grouping #3 (i.e., instances B, D, G, H, I), the fine-grained induced senses would still have a direct correspondence to a single gold-standard sense, which enable a correct prediction of the gold-standard sense from the overly fine-grained induced senses.[3] Hence, a WSI system may produce clusters that are subsets of gold-standard clusters and have high performance on WSID evaluations, even though its clusters are not similar to the gold standard, giving it low performance on WSI clustering comparison evaluations.

In the simple example shown in Figure 6.2, visual inspection provides a reasonable measure of the degree of correspondence; however, computational methods are needed to quantitatively measure correspondence. For WSI systems, evaluations have incorporated multiple types of methods for measuring the correspondence between the gold-standard and induced sense clusterings:

- SemEval-2007 (Agirre and Soroa, 2007) used the FScore measure of Zhao and

[3]We note that in the WSID setups used in Chapter 3, clusters must have at least two instances, with one used for training and the others for testing.



(a) The gold standard grouping of instances by sense

(b) Induced sense instance grouping #1

(c) Induced sense instance grouping #2

(d) Induced sense instance grouping #3

Figure 6.2: Examples of instances (represented as white circles labeled by letters) clustered according to a gold standard (6.2a) and by induced senses (6.2b–6.2d).

Karypis (2005);

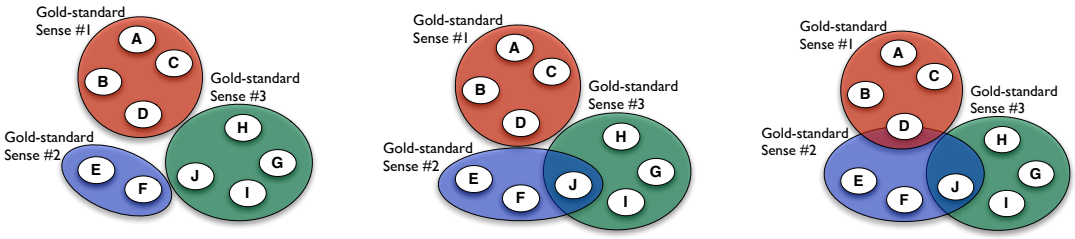
- SemEval-2010 (Manandhar et al., 2010) included the V-Measure (Rosenberg and Hirschberg, 2007), paired FScore (Artiles et al., 2009) and then later reported results for B-Cubed (Amigó et al., 2009) and Adjusted Mutual Information (Vinh et al., 2010); and

- SemEval-2013 Task 11 (Navigli and Vannella, 2013) used the Rand Index (Rand, 1971), Adjusted Rand Index (Hubert and Arabie, 1985), F1 measure (van Rijsbergen, 1979), and Jaccard Index (Jaccard, 1901).

Despite their variety, these clustering comparison measures fall into two general categories: (1) item-based comparison and (2) cluster-based comparisons. Item-based comparisons such as B-Cubed and the paired FScore measure cluster correspondence based on similarities in item's cluster memberships in the two groupings. For example, the paired FScore measures the number of item pairs in a cluster in Grouping A that also appear together in a cluster in Grouping B. In contrast, cluster-based evaluations, such as the V-Measure and Adjusted Mutual Information, measure the degree of correspondence between clusters in each group as a whole. For example, the V-Measure tests the degree to which instances in each cluster in Grouping A are members of the same cluster in Grouping B (along with the reverse B-to-A comparison). We refer the interested reader to Albatineh et al. (2006) and Meilă (2007) for reviews of cluster comparison methods.

In general, the two types of measures provide complementary information about the degree of correspondence between the WSI-based and gold-standard clusters. Because item-based measures test correspondence by the co-occurrence of items or pairs in clusters, the alignment of larger clusters accounts for most of the degree of correspondence by virtue of them having more items or pairs. In contrast, cluster-based methods measure correspondence between clusters, and therefore clusters of all sizes contribute equally to the degree of correspondence value. In this sense, the two measure serve as approximations of macro-averaging, where performance is averaged across classes (i.e., senses) and classes are treated equally, and micro-averaging, where performance is averaged across instances.

In SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013), WSI systems were evaluated on instances that may be labeled with (1) multiple senses and (2) with weights for each sense, according to its applicability in the given context. This introduces two issues with current cluster evaluation methods used in WSI evaluation. First, the current comparison methods are only designed for comparing partitions of instances (also known as hard clusterings), where each instance is assigned to only one cluster. Sec-



(a) A hard clustering, where an instance may be in only one cluster

(b) A soft clustering, where an instance may be a full member of multiple clusters

(c) A fuzzy clustering where an instance may be a partial member of multiple clusters

Figure 6.3: Comparisons between hard (6.3a), soft 6.3b, and fuzzy clusterings 6.3c of instances, depicted as white lettered circles.

ond, because an instance's senses are labeled with weights, an instance may have only a *partial* membership in a cluster. Figure 6.3 illustrates these two types of grouping with respect to sense assignment. Figure 6.3a shows ten instances grouped into three senses, where each instance is labeled with one sense. In Figure 6.3b, instance J has been labeled with two senses and is a therefore a member of two clusters. SemEval-2013 Task 13 allows for weights on sense assignments, which corresponds to Figure 6.3c in which (1) J is a member of two clusters because the instance was labeled with two senses that were fully applicable, while (2) D is a full member of the cluster for Sense #1 but a *partial* member of the cluster for Sense #2, because Sense #2 was not weighted as fully applicable for instance D. Following convention, we refer to partial clustering memberships as fuzzy memberships and a grouping of instances with fuzzy memberships as a *fuzzy clustering*.[4]

To address this limitation of current evaluation methods, we propose extensions to two prior methods, B-Cubed and Normalized Mutual Information (NMI), to evaluate WSI systems producing fuzzy clusterings. B-Cubed and NMI were selected to provide

[4]We note that in the general fuzzy clustering setting, an instance could potentially not be a full member of any cluster (i.e., have all of its memberships be weighted less than one). However, in our sense annotated data set, at least one sense is fully-applicable for an instance, and therefore an instance will always be a full member of at least one cluster.

item-based and class-based evaluations, respectively, based on their prior performance in WSI tasks. We describe the extensions to each method next in Sections 6.2.1.1 and 6.2.1.2.

6.2.1.1 Fuzzy B-Cubed

Bagga and Baldwin (1998) proposed a clustering evaluation known as B-Cubed, which compares two clusterings by examining the similarity of cluster memberships for each instance, rather than each class. Amigó et al. (2009) later extended the definition of B-Cubed to compare overlapping clusters. We generalize B-Cubed further to handle the case of fuzzy clusters.

B-Cubed is based on precision and recall, which estimate the fit from clustering X to Y and Y to X, respectively. For an instance i, precision reflects how many instances assigned to the same sense as i in X are also to the same sense in Y; conversely, recall measures how many instances assigned to the same sense as i in Y also have the same sense assignment in X. The final B-Cubed value is the harmonic mean of the two scores.

To generalize B-Cubed to fuzzy clustering, we adopt the formalization of Amigó et al. (2009), who define item-based precision and recall functions, P and R, in terms of a correctness function, $C \rightarrow \{0, 1\}$. For notational brevity, let avg be a function that returns the mean value of a series, and $\mu_x(i)$ denote the set of clusters in clustering X of which item i is a member (i.e., the senses assignments of i). B-Cubed precision and recall may therefore calculated over all n items as:

$$\text{B-Cubed Precision} = \underset{i}{\text{avg}}[\underset{j \neq i \in \cup\mu_y(i)}{\text{avg}} P(i,j)] \tag{6.1}$$

and

$$\text{B-Cubed Recall} = \underset{i}{\text{avg}}[\underset{j \neq i \in \cup\mu_x(i)}{\text{avg}} R(i,j)]. \tag{6.2}$$

When comparing clusterings where an instance is assigned to at most one sense, P and R are defined as 1 if i and j are assigned to the same sense and 0 otherwise. For

overlapping clusters, Amigó et al. (2009) redefine P and R using *set*-based operations to achieve the same objective as the original definition: When a pair i, j are assigned to more senses in X than in Y, precision decreases; conversely, when the pair are assigned to more senses in Y than in X, recall decreases.

To generalize B-Cubed for fuzzy clustering, we redefine P and R to account for differences in the partial cluster membership of instances. Let $\ell_X(i)$ denote the set of senses to which i is assigned in X, and $w_k(i)$ denote the weight of sense k for instance i. We therefore define the correctness function, C, of two items with respect to X as

$$\mathrm{C}(i, j, X) = \sum_{k \in \ell_X(i) \cup \ell_X(j)} 1 - |w_k(i) - w_k(j)|. \tag{6.3}$$

Equation 6.3 is maximized when i and j are assigned to the same senses in X and have identical weights for those senses. Importantly, Equation 6.3 is equivalent to the original B-Cubed correctness definition of Bagga and Baldwin (1998) when comparing hard clusters and the extended definition of Amigó et al. (2009) when comparing soft clusters. Item-based Precision and Recall are then defined using Equation 6.3 as

$$P(i, j, X) = \frac{\mathrm{Min}(\mathrm{C}(i, j, X), \mathrm{C}(i, j, Y))}{\mathrm{C}(i, j, X)}$$

and

$$R(i, j, X) = \frac{\mathrm{Min}(\mathrm{C}(i, j, X), \mathrm{C}(i, j, Y))}{\mathrm{C}(i, j, Y)}.$$

These generalizations of precision and recall are used in Equations 6.1 and 6.2 to extend B-Cubed for fuzzy clustering.

6.2.1.2 Fuzzy Normalized Mutual Information

Mutual information measures the dependence between two random variables (Danon et al., 2005). In the context of comparing two clusterings X and Y, mutual information reflects the information of the cluster memberships in X provide about the cluster memberships in Y. When X and Y are highly dissimilar, mutual information is minimized, whereas the value is maximized when the two are identical.

Formally, mutual information may be defined as

$$I(X;Y) = H(X) - H(X|Y) \tag{6.4}$$

where $H(X)$ denotes the entropy of the random variable X that represents a particular assignment of n instances into clusters. The entropy of a particular assignment, X, is measured as

$$\begin{aligned} H(X) &= \sum_{x_i \in X} p(x_i) \log_2(p(x_i)) \\ &= \sum_{x_i \in X} \frac{freq(x_i)}{n} \log_2 \left(\frac{freq(x_i)}{n} \right) \end{aligned} \tag{6.5}$$

where $p(x_i)$ denotes the probability of an item being assigned to cluster x_i and $freq(x_i)$ denotes the number of items that were assigned to cluster x_i. Similarly, the conditional entropy of X given Y is computed as

$$\begin{aligned} H(X|Y) &= \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 p(x_i|y_j) \\ &= \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 \frac{p(x_i)}{p(x_i, y_j)} \\ &= \sum_{x_i \in X} \sum_{y_j \in Y} \frac{freq(x_i, y_j)}{n} \log_2 \left(\frac{\frac{freq(x_i)}{n}}{\frac{freq(x_i, y_j)}{n}} \right) \end{aligned} \tag{6.6}$$

Because the value of $I(X;Y)$ varies based on the clusters in X and Y, $I(X;Y)$ is typically normalized into $[0, 1]$ to enable comparing mutual information values from different clustering solutions on the same scale (Luo et al., 2009), with $\max(H(X), H(Y))$ being the recommended normalizing factor (Vinh et al., 2010). Hence, Normalized Mutual Information (NMI) is defined as

$$\text{NMI} = \frac{H(X) - H(X|Y)}{\max(H(X), H(Y))} \tag{6.7}$$

In its original formulation, NMI is defined only for comparing two hard clusterings. Lancichinetti et al. (2009) extend this definition for for calculating NMI between soft clusterings. In the original NMI definition for hard clusterings, each clustering is represented as a discrete random variable whose states denote the probability of an instance being assigned to each cluster. In the overlapping cluster setting, rather than represent the entire clustering as a single variable, Lancichinetti et al. (2009) define each cluster x_i as a separate Bernoulli distribution representing the probability of an instance being

a member of that cluster; i.e., the cluster assignments for each instance is treated as a sample from the variable $\mathbf{X}_{1...k}$ where $\mathbf{X}$ denotes the entire clustering and the ith entry of $\mathbf{X}$ is the Bernoulli distribution for cluster i in $\mathbf{X}$. We denote the distribution of cluster i as X_i.

In the fuzzy clustering setting, an instance's membership in a cluster is no longer binary (i.e., either a member or not), but rather a continuous value in in $[0, 1]$, reflecting the degree of applicability of that cluster's sense to the instance. Therefore, each cluster X_i can be represented separately as a continuous random variable, with the entire fuzzy clustering denoted as the variable $\mathbf{X}_{1...k}$, where the ith entry of $\mathbf{X}$ is the continuous random variable for cluster i. This definition represents cluster membership in the continuous domain, which requires using differential entropy for the continuous variables. However, differential entropy does not have the same behavior as entropy on discrete variables and may have a negative value, which prevents calculating a meaningful value of mutual information to compare fuzzy clusterings.

To compute mutual information for comparing fuzzy clusterings without requiring differential entropy, we propose an alternative setup that discretizes the continuous values of X_i that represent cluster memberships. For the continuous random variable X_i representing the distribution of memberships in cluster i, we discretize the value by dividing the probability mass into discrete bins; that is, the support of X_i is partitioned into disjoint ranges, each of which represents a discrete outcome of X_i that denotes an assignment of an instance to cluster i with the outcome's corresponding membership weight. Because discretization produces a finite number of membership weights, X_i becomes a *categorical distribution* over a set of weights ranges $\{w_1, \ldots, w_n\}$ that denote the weight of an instance's membership in the cluster. With respect to sense annotation, this discretization process is analogous to having an annotator rate the applicability of a sense for an instance using a Likert scale instead of using a rational number within a fixed bound.

Discretizing the continuous cluster membership ratings into bins allows us to avoid

the need for differential entropy (which may be negative) and still extend the definition of mutual information from a binary cluster membership to one of degrees. Using the definition of X_i and Y_j as a categorical variables over discrete ratings, we may then estimate the entropy as follows.

$$H(X_i) = \sum_{i=1}^{n} p(w_i) \log_2 p(w_i) \tag{6.8}$$

where $p(w_i)$ is the probability of an instance being labeled with rating w_i for the sense denoted by X_i. Similarly, we may define the joint entropy of two fuzzy clusters as

$$H(X_k, Y_l) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(w_i, w_j) \log_2 p(w_i, w_j) \tag{6.9}$$

where $p(w_i, w_j)$ is the probability of an instance being labeled with rating w_i in cluster X_k and w_j in cluster Y_l, and m denotes the number of bins for Y_l. The conditional entropy between two clusters may then be calculated as

$$H(X_k|Y_l) = H(X_k, Y_l) - H(Y_l).$$

Together, Equations 6.8 and 6.9 may be used in conjunction with the extension of Lancichinetti et al. (2009) to define $I(X, Y)$ as in the original definition (Eq. 6.4). For brevity, we omit the full details here and present a complete definition in Appendix A. Based on our observations of the expected sense applicabilities in the corpus, we selected uniformly distributed bins in $[0, 1]$ at 0.1 intervals when discretizing an instance's weight for a given cluster. However, we note that other automatic techniques exist for choosing the number and sizes of bins used with discretization (Liu et al., 2002; Kotsiantis and Kanellopoulos, 2006).

6.2.2 WSD Evaluation Measures

Word Sense Disambiguation is traditionally evaluated using precision and recall. Precision is the percentage of disambiguated items that are correct and recall is the percentage of all items that are disambiguated correctly. The two values differ only when a WSD system does not disambiguate all of the instances.

In our work in Jurgens (2012) and for SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013), we propose three objectives to evaluate WSD systems that report multiple senses with respect to human judgments: (1) Detecting all applicable senses in a given context, (2) Ranking all applicable senses by their applicability, and (3) Rating each sense's applicability. Each objective was then matched with a specific measure to quantify the degree to which a WSD system meets the objective. Following, we describe the three measures in Sections 6.2.2.1–6.2.2.3.

6.2.2.1 Jaccard Index

The Jaccard Index is used to measure the degree to which a WSD method identifies all the applicable senses for an instance. Given two sets of senses assigned to an instance, X and Y, the Jaccard Index measures the agreement as $\frac{|X \cap Y|}{|X \cup Y|}$. The Jaccard Index is maximized when X and Y have the same senses, and is minimized when the sets are disjoint, i.e., use completely different senses.

6.2.2.2 Positionally-Weighted Kendall's τ

Positionally-Weighted Kendall's τ is used to measure the degree to which a WSD system can correctly rank senses by their applicability, using the method of Kumar and Vassilvitskii (2010). Given an instance, all senses are ordered by applicability, producing a ranking where each sense occupies a position in the ranking and tied senses may have the same position. In the original, unweighted version, given two rankings of senses, Kendall's τ distance is defined as the number of one-position swaps required to turn one ranking's ordering into the other. In the context of sense annotation, only a few senses are generally applicable, which places a higher emphasis on producing a correct ordering of the most-applicable senses. Therefore, we adopt the positionally-weighted version of Kendall's τ distance, which decreases the distance penalty for differences in the lowest ranks, i.e., the distance between rankings is most affected by differences in

rank among the most-applicable senses.

Formally, Kumar and Vassilvitskii (2010) extend the τ distance definition using a variable penalty function δ for the cost of swapping two positions, which we denote K_δ. By using an appropriate δ, K_δ can be biased towards the correctness of higher ranks by assigning a smaller δ to lower ranks. Because K_δ is a distance measure, its value range will be different depending on the number of ranks used. Therefore, to convert the measure to a similarity we normalize the distance to $[0, 1]$ by dividing by the maximum K_δ distance and then subtracting the distance from one. Given two rankings x and y where x is the reference by which y is to be measured, we may compute the normalized similarity using

$$K_\delta^{\text{sim}} = 1 - \frac{K_\delta(x, y)}{K_\delta^{\text{max}}(x)}. \tag{6.10}$$

Equation 6.10 has its maximal value of one when ranking y is identical to ranking x, and its minimal value of zero when y is in the reverse order as x. We refer to this value as the positionally-weighted Kendall's τ similarity, K_δ^{sim}. As defined, K_δ^{sim} does not account for ties. Therefore, we arbitrarily break ties in a deterministic fashion for both rankings. This deterministic tie breaking ensures that if two items occur in tied rank i in both rankings x and y, they will be assigned rank i and $i - 1$ in both items, thereby preserving the intention of the evaluation measure, i.e., the score is not penalized. Second, we define δ to assign higher cost to the first ranks: the cost to move an item into position i, δ_i, is defined as $\frac{n-(i+1)}{n}$, where n is the number of senses.

6.2.2.3 Weighted Normalized Discounted Cumulative Gain

Weighted Normalized Discounted Cumulative Gain (WNDCG) is used to measure the ability of a WSD system to accurately estimate the applicability weights of senses for a given instance. This measure is an extension of Discounted Cumulative Gain (DCG) which is used in Information Retrieval settings to compare a method's ranking against a gold-standard baseline (Moffat and Zobel, 2008). Given (1) a gold-standard weighting

of the k senses applicable to a context, where w_i denotes the applicability for sense i in the gold standard, and (2) a ranking of the k senses by a WSD system, the DCG may be calculated as

$$\sum_{i=1}^{k} \frac{2^{w_i+1} - 1}{log_2(i+1)}.$$

DCG is commonly normalized to $[0, 1]$ so that the value is comparable when computed on rankings with different numbers of items (k) and weight values. To normalize, the maximum value is calculated by first computing the DCG on the ranking when the k items are sorted by their weights, referred as the Ideal DCG (IDCG), and then normalizing as $\frac{DCG}{IDCG}$ to compute the Normalized Discounted Cumulative Gain (NDCG).

The DCG only takes into account the weights assigned in the gold standard, which potentially masks important differences in the weights assigned to the senses by the WSD system. WSD systems producing the same ranking from their sense weights will have identical DCG values, despite any differences in the systems' weights. Therefore, we propose weighting the DCG by the relative difference between the system's and gold-standard's weights. Given an alternate weighting of the k items by the WSD system, denoted as $\hat{w}_i$,

$$\text{WDCG} = \sum_{i=1}^{k} \frac{\frac{\min(w_i, \hat{w}_i)}{\max(w_i, \hat{w}_i)} \left(2^{w_i+1} - 1\right)}{log_2(i)}. \tag{6.11}$$

The key impact in Equation 6.11 comes from weighting an item's contribution to the score by its relative deviation in absolute weight. A set of weights that achieves an equivalent ranking may have a low WDCG if the weights are significantly higher or lower than the reference. Equation 6.11 may be normalized in the same way as the DCG. We refer to this final normalized measure as the Weighted Normalized Discounted Cumulative Gain (WNDCG).

6.3 Experiment 1: SemEval-2013 Task 13 Evaluation

The first experiment evaluates our all-words (AWCD) and single-word (SWCD) WSI systems described in Chapter 2 on the SemEval-2013 Task 13 data set (Jurgens and Klapaftis, 2013). Task 13 is a lexical sample WSI and WSD task, where system disambiguate multiple instances of a set of target words. Unlike prior WSI-based SemEval tasks (Agirre and Soroa, 2007; Manandhar et al., 2010), Task 13 includes instances with multiple sense annotations, which WSD systems should recognize as such. For comparison, our evaluation includes the systems that originally participated in the task and four additional supervised and unsupervised WSD systems for measuring differences between WSID and WSD.

Three forms of analysis were performed on this task. In the first, we report results for all instances in the data set, examining the ability of systems to recognize multiple senses. Second, we refine the evaluation setup to only the instances of Task 13 that have a single sense annotation. This second analysis allows for a direct comparison with WSD systems, which were not designed to recognize multiple senses. Third, we test the performance impact of using the sense mapping function of Task 13 versus the linear kernel SVM mapping function used in our pseudoword experiments (Ch. 3).

6.3.1 Data

Task 13 focuses on unsupervised learning and therefore includes no sense-annotated training data. To create a reproducible evaluation setting, task participants were asked to create their WSI models using the ukWaC corpus (Baroni et al., 2009). Previous SemEval WSI tasks had provided participants with corpora specific to the task's target terms; in contrast, this task opted to use a large corpus to enable WSI methods that require corpus-wide statistics, e.g., Van de Cruys and Apidianaki (2011) and our collocation-based methods.

Test data was drawn from the Open American National Corpus (Ide and Suderman,

2004, OANC) across a variety of genres and from both the spoken and written portions of the corpus, described earlier in Chapter 5, Section 5.3.1. The data set contains 4664 contexts for 50 terms, with a minimum of 22 and a maximum of 100 contexts per term.

6.3.2 Evaluation Setup

Task 13 includes two cluster evaluation measures for WSI systems: Fuzzy B-Cubed and Fuzzy NMI (Sec. 6.2.1). For WSID and WSD systems, Task 13 uses three evaluation measures: the Jaccard Index, Weighted Kendall's τ, and Weighted NDCG (Sec. 6.2.2).

To create WSID systems, Task 13 uses an extension of the Agirre et al. (2006b) mapping function proposed in Jurgens (2012) which takes into account sense applicability weights in order to produce annotations with multiple weighted senses.[5] We follow the 80/20 setup of Manandhar et al. (2010), where the corpus is randomly divided into five partitions, four of which are used to learn the sense mapping; the sense labels for the held-out partition are then converted and compared with the gold standard. This process is repeated so that each partition is tested once.

6.3.3 Baselines

Task 13 included four baselines based on modeling two different types of WSI and WSD systems each. WSD systems were evaluated against two baselines: (1) **SemCor MFS** which labels each instance with the most frequent sense of that lemma in SemCor and (2) **SemCor Frequency Weighted** (SCFW) baseline, which labels each instance with all of the target lemma's senses that were seen in SemCor, weighted by the frequency with which they were seen. SemCor MFS is a standard baseline that reflects the expected performance of a system without any knowledge of the sense distribution of the test corpus. The SCFW baseline simulates a system that expects all commonly-seen senses, but with applicability proportional to their frequency.

[5]In the third analysis (Sec. 6.3.7), we measure the impact of using the Linear-kernel SVM mapping function, used in the experiments in Chapter 3, instead of this mapping function.

WSI performance was evaluated against two standard baselines: (1) **1c1inst** which labels each instance with a separate induced sense and (2) **All-instances, One sense** which labels all instances with the same induced sense. These two baselines reflect the extremes of different sense granularity, with the first indicating that each usage has a distinct meaning, while the latter indicating all usages have approximately the same meaning.

Because the sense induction baselines each behave consistently for all instances, they perform in distinct ways when used in the WSID setting with the Task 13 mapping function. The 1c1inst baseline always obtains a zero F1 value when used in the WSID setting; because each sense appears only once, when instances are divided for training and test sets, a sense in the test set will have never been seen in training, making it impossible for the mapping function (cf. Section 3.2.1) to produce a reference annotation from the unseen induced sense. The All-instances, One Sense baseline will become transformed into the *average* sense distribution of the training set.

6.3.4 Systems

Experiment 1 includes both WSI and WSD systems in the evaluation, described next.

6.3.4.1 WSI Systems

WSI systems include a system for each of the two methods proposed in this thesis in Chapter 2 and the seven WSI systems submitted as a part of Task 13. We describe our systems configurations next and then the submitted systems.

Our SWCD system (cf. Sec. 2.3.1) constructed each word's graph using the k-neighbors method, where $k = 1000$ based on the method's best performance in Chapter 3.[6] Our AWCD system (cf. Sec. 2.3.2) constructed its graph from the 20,000 most-

[6]Additional testing on $k = \{500, 2000, 3000, 4000, 5000, 10000\}$ show that 1000 provided the best performance for most evaluations with Task 13, corroborating earlier tests with pseudoword tuning data in Chapter 3.

frequent POS-tagged lemmas in the ukWaC, which were restricted to nouns, verbs, adjectives, and adverb lemmas and excluded a list of overly-common lemmas (i.e., stopwords). These 20K lemmas become vertices in the graph and then edges are added to the 50 lemmas having the highest χ^2 with each vertex. This produced a graph with 964,665 edges and 102,251 vertices. Parameter settings for AWCD were constrained by the size of the graph, where including more initial vertices or more neighbors produced graphs proved too computationally expensive to use with the system's community detection method.

Comparison WSI systems include the seven WSI systems submitted by three teams to Task 13. The AI-KU team submitted three WSI systems with different parameter settings of their algorithm, described earlier in Section 2.5.2. The University of Melbourne (**Unimelb**) team submitted two WSI systems based on the topic-modeling approach of Lau et al. (2012), also described earlier in Section 2.5.2.

The University of Sussex (**UoS**) team (Hope and Keller, 2013b) submitted two WSI systems that use the MaxMax algorithm (Hope and Keller, 2013a). The UoS system first constructs an undirected weighted graph of collocation using Normalized Pointwise Mutual Information (NPMI) (Bouma, 2009). UoS then converts this graph into a directed graph using the weights of the graph: Given two vertices, u and v, a directed edge is added from u to v if the weight of the edge from u to v is maximal for all edges incident on u. This directed graph is then converted into clusters by (1) identifying all the root nodes in the directed graph and (2) putting a root node and all of its descendants into a cluster. The clustering procedure has a tendency to generate many small clusters, so an additional procedure is used to merge clusters containing similar vertices, as measured using NPMI.

6.3.4.2 WSD Systems

Two types of WSD systems are included: (1) the two systems submitted by the La Sapienza team as a part of Task 13, which report multiple senses per instance and (2) four additional publicly-available WSD systems, which report only a single sense. We refer to the second group of WSD system as *comparison WSD systems* to distinguish them from the La Sapienza systems. All publicly-available systems were applied using their standard, off-the-shelf parameter settings. We describe the systems next.

The **La Sapienza** team submitted two unsupervised WSD systems based applying Personalized PageRank (PPR) (Agirre and Soroa, 2009) over a WordNet-based network. Using the method of Pilehvar et al. (2013), probability distributions are created over the network's nodes for (1) each sense of the target word to be disambiguated and (2) the context containing the target word. These probability distributions reflect the relatedness of each node (i.e., sense) in the network to the sense or context, respectively. The final sense annotation is computed by measuring the similarity of the distribution of each of the target word's senses with the context's distribution and reporting all senses with non-zero similarity, weighted according to their similarity.

The **UKB** system of Agirre and Soroa (2009) is an unsupervised graph-based WSD system that uses the WordNet lexical network to perform all-words WSD. For a given context, the senses of the context words are used to create a subgraph in an enriched version of the WordNet network, which contains additional relations, such as those from glosses. Personalized PageRank (Brin and Page, 1998) is then run, using the vertices for the context's senses to seed the initialization vector. After PageRank converges, each lemma is assigned to its sense that had the highest PageRank value.

The **WordNet::SenseRelate** package of Pedersen and Kolhatkar (2009) contains multiple unsupervised algorithms for WSD. We evaluate using the AllWords method, which looks at all content words in a sentence. Each word is disambiguated by comparing its senses to those of content words in a window of $\pm$ three words.

		WSD F1			Cluster Comparison			
Team	System	Jac. Ind.	K_{δ}^{sim}	WNDCG	Fuzzy NMI	Fuzzy B-Cubed	#I	#S
this work	SWCD	0.195	0.603	0.339	**0.085**	0.551	1.67	6.52
this work	AWCD	0.198	0.611	0.288	0.005	**0.616**	1.00	6.58
AI-KU	base	0.196	0.615	**0.387**	0.065	0.390	7.76	6.42
AI-KU	add1000	0.196	0.602	0.216	0.035	0.320	7.76	6.42
AI-KU	remove5-add1000	0.244	**0.639**	0.332	0.039	0.451	3.12	5.21
Unimelb	5p	0.218	0.609	0.364	0.056	0.459	2.37	5.78
Unimelb	50k	0.213	0.615	0.369	0.060	0.483	2.48	5.92
UoS	#WN Senses	0.191	0.592	0.314	0.047	0.201	8.08	6.57
UoS	top-3	0.231	0.621	0.374	0.045	0.448	3.00	5.29
La Sapienza	system-1	0.147	0.508	0.312	-	-	-	8.69
La Sapienza	system-2	0.147	0.510	0.383	-	-	-	8.67
BabelNet-WSD		0.242	0.252	0.180	-	-	-	1.00
IMS		**0.466**	0.471	0.347	-	-	-	1.00
SenseRelate		0.329	0.342	0.245	-	-	-	1.00
UKB		0.185	0.204	0.139	-	-	-	1.00
All-instances, One sense		0.192	0.609	0.288	0.0	0.620	1.00	6.62
1c1inst		0.0	0.0	0.0	0.070	0.0	1.00	0.0
SemCor MFS		0.456	0.464	0.340	-	-	-	1.00
SemCor Sense Frequencies		0.203	0.584	0.413	-	-	-	6.11

Table 6.1: Performance on the five evaluation measures for all system and baselines. Top system performances are marked in bold.

The unsupervised graph-based algorithm of Navigli and Ponzetto (2012b) disambiguates using the BabelNet semantic network (Navigli and Ponzetto, 2012a), a wide-coverage encyclopedic dictionary built by merging WordNet with Wikipedia. We refer to this method as **BabelNet-WSD**. The algorithm disambiguates using the Edge heuristic of Navigli and Lapata (2010) for graph-based WSD in BabelNet, which examines the connectivity of all the synsets of the content words in the context within the larger BabelNet graph.

Last, we include the It Make Sense (**IMS**) system of Zhong and Ng (2010), which was also used in the pseudoword experiments in Chapter 3. Unlike previous WSD systems, IMS is supervised and was trained to perform WSD using the SemCor (Landes et al., 1998) and DSO (Ng and Lee, 1996) sense-tagged corpora.

6.3.5 Analysis 1: All Instances

Table 6.1 shows the results of all systems in their WSD performance on the full data set. In addition to the WSI and WSD measures, we report the average number of induced senses a WSI system used to label each instance as #I and the average number of resulting WordNet senses produced by the corresponding WSID system for each instance as #S. Note that #S is determined by the mapping function for WSID systems and that WSD systems directly produce WordNet senses, so #I is not applicable. In what follows, we first discuss the performances of all systems on WSD evaluations and then discuss their performances on WSI evaluations.

WSD Performance Detecting multiple senses, measured using the Jaccard Index, was the most difficult task both for system originally participating in Task 13 and for our AWCD and SWCD systems. For WSID systems, the mapping function contributed to the low-performance in detecting multiple senses. Nearly all WSI models produced multiple induced senses for each instance, as seen in the #I column of Table 6.1. These induced senses did not have a direct correspondence to WordNet senses and therefore the mapping function in turn produced multiple WordNet senses for each instance, as seen in the #S column. As a result, WSID systems consistently report too many senses and are heavily penalized by the Jaccard Index. We also note that the La Sapienza WSD system, which participated in Task 13, also predicted too many senses and was similarly penalized.

The comparison WSD systems, shown in the middle group of Table 6.1, performed much better at the Jaccard Index evaluation due to their original design of reporting only a single sense per instance. Because 89% of the instances in the test data have only a single sense annotation, the comparison WSD systems benefit from their conservative design. However, most of the comparison WSD systems themselves are not accurate at selecting the correct sense and only the supervised IMS system is able to surpass the MFS baseline.

The measures for ranking senses by applicability (K_{δ}^{sim}) and quantifying their applicability (WNDCG) provide more insight in the different systems' performance. Both measures consider the applicability weights assigned to each sense, so WSID and WSD systems that correctly identify the most-applicable senses will not be penalized as much as with the Jaccard Index if they incorrectly report additional senses that are assigned a low weight. Indeed, the performance of WSID systems and the La Sapienza WSD system (which reports multiple senses) increases significantly for the ranking and quantifying evaluations, relative to that of the comparison WSD systems. Although the WSID and La Sapienza systems reported more senses than were present, their relative ordering and weighting most closely matched that of the gold standard, with many inapplicable senses having low weights. The comparison WSD systems performed relatively poorly at ranking and quantifying sense applicability because their single-sense annotation of an instance was often incorrect. Of the comparison WSD systems, only the supervised IMS system achieves performance comparable to the semi-supervised WSID system; the higher performance of the IMS system is due to it correctly identifying the most-applicable sense.

Examining only the WSID systems, no one WSID system performs best on all metrics. The AI-KU systems perform strongly, though the team's different configurations reveal that the algorithm is susceptible to lower performance with certain parameters (i.e., the "add1000" system). Similarly, the systems of the Unimelb team perform well, despite the simple approach of inducing senses using an off-the-shelf topic modeling algorithm. Our AWCD and SWCD systems perform competitively but do not attain state-of-the-art performance in the WSD evaluations, though the SWCD system does outperform all systems in the WSI evaluations (discussed below). All WSID systems were competitive against the two WSI baselines, except in the case of detecting multiple sense, where the WSID systems reported too many instances. The higher performance of WSID systems at ranking and quantifying sense applicability indicates that the systems are accurate at identifying the most-applicable senses. Hence, performance could

potentially be increased in future systems by suppressing low-weighted senses from being reported in WSID output.

WSI Performance Clustering comparison tests a WSI system's capability at producing sense distinctions similar to those of lexicographers' in the gold standard, which differs from the capability tested in the WSID evaluation, discussed above. Though systems may do well in both types of evaluation, the results of our clustering comparison evaluation reveal that WSI systems differ in their capabilities on the two tasks, with the systems performing best at clustering comparison differing from those best for WSID. Figure 6.4 visualizes the WSI systems' performances on both clustering comparison metrics, also reported in Table 6.1, where systems increase in performance as they move up and to the right. The two WSI baselines, All Instances, One Sense (AIOS) and 1c1inst, appear on the axes as they reflect the extreme behaviors of an WSI system, respectively: (1) labeling all instances with the same induced sense or (2) labeling each instance with a different sense. WSI systems which produce coarse-grained senses whose clusters combine the instances of multiple gold-standard senses will tend towards the performance of the AIOS baseline; in contrast, WSI systems that produce finer-grained senses that subdivide the clusters for gold-standard senses will tend towards the performance of 1c1inst.

Examining the performance of the WSI systems, our SWCD system attains the highest Fuzzy NMI and second-highest Fuzzy B-Cubed scores, demonstrating that the system comes closest to matching human lexicographers' groupings of instances without being biased towards the performances of the baselines. Indeed, the SWCD system is the only system to outperform the 1c1inst baseline on Fuzzy NMI. In contrast, the systems that performed better for WSID, such as the AI-KU systems, attain lesser scores on the clustering evaluations. Examining the number of induced senses produced per instance (Table 6.1, column #I) shows that systems producing large numbers of induced senses per instance perform well for WSID but worse on these two cluster

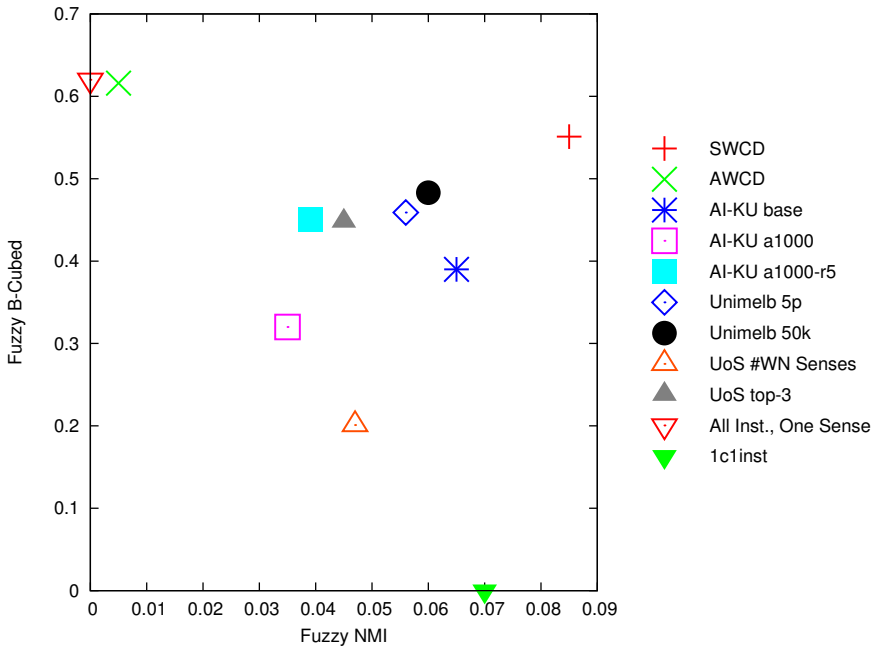


Figure 6.4: Performance of WSI systems according to two clustering comparison measures on all instances of Task 13

comparison measures, where as SWCD and the Unimelb system produce fewer senses per instance and score higher on the clustering comparison measures.

Our AWCD method produces a single sense for nearly all instances; though the AWCD method uses multiple induced senses, these become sense-mapped to the same WordNet sense, causing the system to perform similarly to the All Instances, One Sense baseline. A further analysis of the system revealed that AWCD produced highly-accurate induced senses, represented as clusters of collocations; however, the clusters contained only a tens of collocations, which limited the number of contextual features that the system could match with its senses' features when disambiguating. The inability to match contextual features resulted in the algorithm frequently using its back-off strategy of labeling an instance with the sense having the most features. This lower performance supports our earlier hypothesis in Chapter 2 Section 2.4 that suggested AWCD could have reduced disambiguation capabilities because of the smallness of its clusters. The size of the clusters and subsequent lower performance was largely due to

the computational complexity of community detection method used by AWCD, Ahn et al. (2010), which limited the size of the graph from which collocations clusters could be found. However, this computational limitation raises the possibility that performance might be improved with future, more-scalable community detection methods that could be run on larger graphs in order to produce clusters with more collocations.

This difference in performance between WSI-based evaluations and WSID evaluations suggests that future researchers may be better served by choosing the WSI system that matches their intended use case: lexicographers looking to use a WSI system to group contexts by senses (e.g., the KWIC in Chapter 2) may prefer to use a system such as SWCD, while researchers building a WSID system may prefer to use a system such as AI-KU that offers better features for learning a mapping to a reference sense inventory.

6.3.6 Analysis 2: Single-sense Instances

In the second analysis, we restrict the data to only the subset of instances labeled with single sense. Furthermore, we modified the test setting to have systems also label instances with a single sense: (1) the sense mapping function for WSI systems (Jurgens, 2012) was modified so that after the mapping, only the highest-weighted WordNet sense was reported, and (2) the La Sapienza system output was modified to report only the highest-weighted sense from its annotation. In the single-sense setting, the mapping function converts the All Instances, One Sense baseline into the most-frequent sense in the test data (compared with that in SemCor). WSID Systems are evaluated using the standard WSD Precision and Recall measures (Sec. 6.2.2) and we report the F1 measure of Precision and Recall. WSI systems are evaluated using the original definitions of B-Cubed and NMI for hard clusterings. This second analysis removes the additional difficulty of recognizing multiple senses and also enables directly comparing WSID methods with the WSD methods that had been designed to produce only a single sense.

Team	System	F1	NMI	B-Cubed
this work	SWCD	0.577	**0.065**	0.499
this work	AWCD	0.570	0.004	**0.563**
AI-KU	base	**0.641**	0.045	0.351
AI-KU	add1000	0.601	0.023	0.288
AI-KU	remove5-add1000	0.629	0.026	0.421
Unimelb	5p	0.596	0.035	0.421
Unimelb	50k	0.605	0.039	0.441
UoS	#WN Senses	0.574	0.031	0.180
UoS	top-3	0.600	0.028	0.414
La Sapienza	System-1	0.204	-	-
La Sapienza	System-2	0.217	-	-
BabelNet-WSD		0.256	-	-
IMS		0.491	-	-
SenseRelate		0.338	-	-
UKB		0.189	-	-
All-instances, One sense		0.569	0.0	0.570
1c1inst		0.0	0.048	0.0
SemCor MFS		0.477	-	-

Table 6.2: System performance in the single-sense setting. Top system performances are marked in bold.

Table 6.2 shows the systems' performance on single-sense instances, revealing three main insights. First, WSID systems have a substantial increase in performance and improvement over the MFS baseline compared with the multiple sense settings. All of the WSID systems surpass the performance of every unsupervised and supervised WSD system we tested and surpassed the average performance of many supervised WSD systems tested in previous WSD evaluations (Kilgarriff, 2002; Mihalcea et al., 2004; Pradhan et al., 2007; Agirre et al., 2010). Similar to the results of the pseudoword experiment in in Chapter 3 Section 3.5, the high performance of the WSID systems suggest that the combination of a small amount of sense-annotated data with a WSI model trained on a large amount of data can produce a semi-supervised WSID system that outperforms current WSD state of the art. Indeed, WSID systems were trained with

an average of 66.0 instances per lemma, which is within the range of training data sizes where WSID systems outperformed the supervised WSD system in the pseudoword experiments.[7] Among the WSD systems, the supervised IMS system performed best, with the unsupervised approaches performing less well. However, all WSID based systems outperform IMS by at least 0.079.

Second, among the WSID and WSD systems, the distributional systems that use positional or syntactic information as features (AI-KU, Unimelb, SenseRelate, and IMS) perform better than graph-based approaches (AWCD, SWCD, UoS, La Sapienza, BabelNet-WSD, UKB), which only use collocation features. The AI-KU and Unimelb WSID systems and the SenseRelate and IMS WSD systems all use information on which lemmas appear in the immediate context of the target word being disambiguated, while the remaining graph-based systems ignore syntactic information and simply use the occurrences of lemmas anywhere in the context. Although the distributional approaches vary in how the positional and syntactic features are used, the performance difference between distributional and graph-based approaches suggests that syntactic and positional information is important for attaining high WSD performance and that future graph-based approaches would be served well by incorporating it into their features.

Third, similar to what was observed in the full data set, WSI systems vary in their performances based on the evaluation, with our SWCD method attaining superior performance on the WSI-based clustering comparison evaluations, while the AI-KU systems attain superior performance on the WSID evaluations. Despite these differences, all systems were able to surpass the SemCor MFS baseline and most systems were able to surpass the more-competitive All Instances, One Sense baseline,[8] demonstrating that

[7] We note that in the pseudoword experiments, the supervised IMS system and WSID systems were trained on the same sense-annotated data. In contrast, in the current evaluation, WSID systems and IMS used different training data: the WSID systems were trained as a part of the cross-validation procedure on the Task 13 data set while IMS was used with its off-the-shelf trained models which incorporate significantly more training data from multiple corpora.

[8] As noted earlier, the All Instances, One Sense baseline is equivalent to the most frequent sense in the training data and represents the best performance that any system could obtain if a single sense is used

Team	System	F1 per mapping function		Performance Difference
		Jurgens (2012)	Linear-kernel SVM	
	SWCD	0.577	0.577	0.000
	AWCD	0.570	0.570	0.000
AI-KU	base	0.641	0.634	-0.007
AI-KU	add1000	0.601	0.601	0.000
AI-KU	remove5-add1000	0.628	0.635	+0.007
Unimelb	5p	0.596	0.600	+0.004
Unimelb	50k	0.605	0.603	-0.002
UoS	#WN Senses	0.574	0.599	+0.025
UoS	top-3	0.600	0.604	+0.004

Table 6.3: WSD performance on single-sense instances where WSID systems were built using either the mapping function of Jurgens (2012) or the linear-kernel SVM described in Section 3.2.1.

the WSID systems are identifying salient distinctions in the word's usages.

6.3.7 Analysis 3: Impact of the Mapping Function

The third analysis measures the impact of the sense mapping function on WSID performance in Task 13. Task 13 uses the mapping function of Jurgens (2012), which extended the method of Agirre et al. (2006b) for producing sense annotations with multiple senses. However, in Chapter 3, we demonstrated that a linear-kernel SVM produces superior performance to the method of Agirre et al. (2006b) on data annotated with a single sense. Therefore, to test the difference between the two, we repeat the single-sense setup of the previous analysis (Sec. 6.3.6) and measure performance of each system with the two mapping functions.

Table 6.3 shows the F1 score of each WSID system with both mapping functions and in the third column shows the performance difference. Consistent with the pseudoword experiments in Chapter 3, the results here show that using a linear kernel SVM for a mapping function improves performance for the majority of systems. System F1

for all instances of a lemma. In this setting, the baseline could be thought of as a supervised baseline with knowledge of the expected sense distribution in the test data.

scores increase by an average of 0.005, though the top-performing system, AI-KU base, does decrease in performance. Labeling instances with only a single sense is most often the correct behavior for a WSD system (as shown in Chapter 5) and therefore the linear kernel SVM could be considered as an effective mapping function for most instances, given that a usage may only rarely have multiple senses.

6.4 Experiment 2: Ensemble WSID Performance

The second experiment evaluates the ensemble WSID setup described in Chapter 3 Section 3.2.3 on real-world sense-annotated data instead of pseudowords. This experiment is test the performance of ensembles built from a variety of WSI models using multiple sense inventories.

6.4.1 Experimental Setup

To evaluate the ensemble WSID setup with sense-annotated data, we use the three SemEval tasks have included WSID evaluation: SemEval-2007 Task 2 (Agirre and Soroa, 2007), SemEval-2010 Task 10 (Manandhar et al., 2010), and SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013). We repeat each task's exact evaluation setup, with the exception of substituting the task's mapping function with a linear kernel SVM when constructing the ensemble.

Two significant differences exist in the tasks' setups compared with the pseudoword experiments in Chapter 3. The 2007 and 2010 tasks use OntoNotes senses (Hovy et al., 2006), which are known to be more coarse-grained than the WordNet senses which the pseudowords approximate. Second, the 2013 task evaluates models when the annotation is potentially ambiguous and includes gold standard data where the instances have multiple sense labels. Because we consider only instances labeled with a single sense in the pseudoword experiments, we restrict our analysis of the 2013 task to only those instances with a single sense label.

Ensemble systems were created using the induced sense answers from the systems that participated in each task and a linear kernel SVM to perform the sense mapping. We intentionally use the original WSI models rather than the five models used in our earlier experiments in order to test the benefits of the proposed WSID configuration with a wider variety of systems. For each task, we consider two ensembles: (1) the outputs of all WSI systems, and (2) the outputs of the best configuration of each system, measured according to their WSID performance in the original task. We note that the 2007 task allowed only one configuration per system, so only one ensemble is produced for that task. Additionally, we exclude the Duluth-R systems' results in the 2010 task because these systems were intentionally submitted as random baselines.

6.4.2 Results

Table 6.4 shows the ensembles' performances in all three tasks and includes the scores of best-performing system in each task originally and the task's MFS baseline. In each task, our ensemble WSID configuration shows performance improvements over the best-performing system and MFS. Improvements over MFS and the best-performing system were all significant at $p < 0.01$ using McNemar's test for significance. We also note that all ensemble performance improvements over the best-performing system are larger than the differences between the performances of the best and second-best systems in each task. These results demonstrate consistent performance improvements for an SVM-based ensemble WSID model even when using different sense inventories and entirely different sets of WSI systems.

6.5 Conclusion

This chapter has presented a quantitative evaluation of WSI, WSID, and WSD systems, including the WSI systems presented in Chapter 2. In our first experiment, we test systems on the SemEval-2013 Task 13 data set (Jurgens and Klapaftis, 2013), which

			Ensembles	
SemEval	MFS	Best System	All-Systems	Best-Configuration
2007	0.787	0.816	**0.828**	-
2010	0.587	0.624	**0.680**	0.670
2013	0.477	0.641	0.644	**0.657**

Table 6.4: A comparison of the best-performing system in each SemEval WSID task and our proposed ensemble method.

includes instances labeled with multiple senses and perform three analyses. In the first analysis, we demonstrate that WSID systems offer superior performance to unsupervised and supervised WSD systems and surpass the competitive MFS baseline. Furthermore, we show that our SWCD WSI system achieves state-of-the-art performance at matching gold-standard groupings of instances according to their WordNet senses, providing a valuable tool for assisting lexicographers. In the second analysis, we evaluate on only the single-sense instances of Task 13 and demonstrate that WSID systems outperform both supervised and unsupervised WSD systems in this setting. The results of the second analysis show that in cases of limited training data, WSID can achieve superior performance even to a supervised system trained on more sense data, further confirming the pseudowords results in Chapter 3 with performance on real-world data. In the third analysis, we test the impact of the sense-mapping function on real-world data, showing that WSID systems using a linear kernel SVM mapping function improves the performance of most WSID systems. This third result suggests that the linear kernel SVM should be adopted as the standard sense mapping function both for WSID systems and SemEval-like evaluation setups.

The second experiment tested our procedure from Chapter 3 for constructing ensemble WSID systems. Using multiple ensembles constructed from the systems of three SemEval WSI tasks (Agirre and Soroa, 2007; Manandhar et al., 2010; Jurgens and Klapaftis, 2013), we demonstrate that in all tasks, our ensemble WSID procedure provided a statistically significant performance improvements over the best system and

MFS baseline for each task.

CHAPTER 7

Conclusion

7.1 Summary and Contributions

Word Sense Disambiguation (WSD) is an important but notoriously challenging task in NLP where the particular meaning of a word in context must be identified (Ide and Véronis, 1998; Navigli, 2009). Two key aspects of WSD increase its difficulty. First, due to the time and costs required to produce high-quality sense-annotated corpora, currently-available sense annotated data is sparse, often with only a few hundred examples at most for any give meaning of a word. Second, the different meanings of a word may be related, which increases the difficulty of distinguishing between such senses, especially given that related senses may appear in similar types of contexts (Palmer et al., 2007).

This dissertation has addressed these two challenges in three ways. First, we propose new unsupervised WSI methods (Ch. 2 and (Jurgens, 2012)) and semi-supervised WSID methods (Ch. 3) for making better use of the limited sense-annotated data. Towards this first goal, we provide three contributions. First, we introduce two novel graph-based WSI methods, which we demonstrate as identifying meaningful senses and providing superior performance to other graph-based WSI approaches. In particular, our SWCD method achieves new state-of-the-art performance, surpassing all types of WSI systems at matching gold-standard instance groupings in SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013). Second, we proposed a new method for producing semi-supervised WSID systems and demonstrate that our new method produces a

state-of-the-art WSID performance on multiple benchmarks. Furthermore, as a part of our WSID analysis, we produced a new pseudoword evaluation, which closely approximates WSD performance on polysemous WordNet nouns. Our new evaluation provides two orders of magnitude more data and precise controls for the frequency of senses, which enables testing WSID systems with varying amounts of data and sense distributions. Third, using our novel pseudoword evaluation, we quantify the performance differences between WSID and fully supervised WSD, showing that our novel WSID ensemble setup attains superior performance to a state-of-the-art supervised WSD system when limited sense-annotated data is available. Our third contribution enables experimenters to choose the appropriate learning algorithm (WSID or supervised WSD) given the amount of training data available to them.

In the second focus of the dissertation, we propose to reduce data sparsity by new methods for gathering high-quality sense annotations at a reduced cost from non-experts. We propose and evaluate new annotation procedures for use in crowdsourcing (Ch. 4 and Jurgens (2013)). In our analysis of prior methods and our novel annotation methods, we demonstrate that our MaxDiff-based annotation procedure enables untrained workers to produce sense annotations that are highly replicable and achieve agreement rates on part with those seen in expert-based corpora. Furthermore, our analysis shows that although workers had difficulties with the traditional annotation method of requiring one sense per instance, workers were much more accurate when the task allowed them to express their uncertainty about the correct annotation by marking all the senses that they thought were appropriate.

In the third focus, we analyze cases of ambiguity in annotation in order to identify the causes behind the difficulty and how future efforts may reduce their errors (Ch. 5 and Jurgens (2014)). As a part of this analysis, we (1) generated the largest corpus of instances annotated with all applicable WordNet senses, weighted according to the senses' applicabilities and (2) provided new evaluation measures for testing WSI and WSD systems on multiple-sense instances. In our analysis, we demonstrate that most

ambiguity is due to the usage's containing sentence having too few semantic cues; however, a sizable minority of instances are ambiguous due to syllepsis, where contextual cues have elicited different senses. Further, we demonstrate that using a coarse-grained sense inventory such as OntoNotes (Hovy et al., 2006) can remove the need for using multiple sense in most cases – though nearly a quarter of ambiguous noun instances would still need multiple coarse-grained senses.

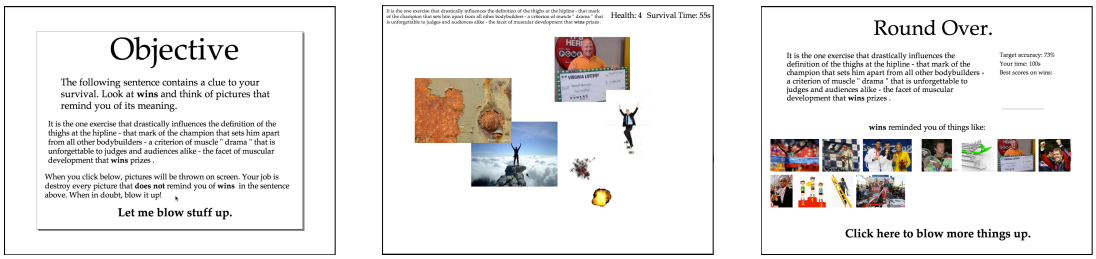
7.2 Future Work

The research in this dissertation has inspired several other currently-ongoing research projects as future work, three of which we highlight here.

7.2.1 Sense Annotating via Video Games with a Purpose

Crowdsourcing provide one means of gathering new sense annotated data for WSD. However, crowdsourcing still costs money, which limits the amount of sense annotated data that can be produced based on the available financial resources. An alternate method of gathering sense annotated data is to use Games with a Purpose, where a task (such as sense annotation) is turned into a game objective. Players play the game and as a result complete the annotation task implicitly, often without being aware of the annotation task. If the game is sufficiently fun in its own right, players may continue to play for free, removing the financial limitations on the amount of data that can be produced.

Within NLP, several works have proposed using games for tasks such as anaphora resolution (Hladká et al., 2009; Poesio et al., 2013), paraphrasing (Chklovski and Gil, 2005), term associations (Artignan et al., 2009), and WSD (Seemakurty et al., 2010; Venhuizen et al., 2013). Notably, all of these linguistic games focus on users interacting with text, in contrast to other highly successful games with a purpose such as Foldit (Cooper et al., 2010), in which players fold protein sequences, and the ESP game (von



(a) The context and target word are shown to the player before the start of gameplay

(b) Players destroy image by clicking or touching, which annotates the usage.

(c) The round-over summary provides feedback to the players on their accuracy.

Figure 7.1: Screenshots of the three key elements of the Ka-boom! video game, which performs word sense annotation.

Ahn and Dabbish, 2004), where players label images.

Two prior works have investigated using games to perform WSD: Wordrobe (Venhuizen et al., 2013) and Jinx (Seemakurty et al., 2010). Wordrobe asks players to disambiguate nouns and verbs using multiple choice questions where each option is a sense definition, making the game appear very similar to a standard sense annotation task and likely less appealing to players. Jinx uses two players who both have to independently provide lexical substitutes of an ambiguous word and are then scored on the basis of their shared substitutes. Jinx has a more game-like feel than Wordrobe. However, the substitutes do not directly produce sense annotations, which must be generated from a heuristic search based on the locality of the substitutes in the WordNet hypernym graph.

In our ongoing work, we have proposed a radical shift in game design for NLP and propose annotating using *video games*, where players engage in graphical environments and play games resembling those on common video game platforms. In our initial games, we have produced two video games to construct a mapping between words and WordNet senses (Vannella et al., 2014). Further, we produced a third game, *Ka-boom!*, that produces fine-grained sense annotations as a result of image-based game play.

Ka-boom! is an action game in the style of the popular Fruit Ninja game: pictures

are tossed on screen from the boundaries of the screen, which the player must then selectively destroy in order to score points. Prior to the start of a round of game play, players are shown a sentence with a word in bold (Fig. 7.1a) and asked to envision pictures related to that word's meaning in the context. Players are then instructed to destroy pictures that do not remind them of the meaning of the word in bold (i.e., to allow all the pictures showing the word's concept to survive). Following, players begin a round of game play, seen in Figure 7.1b, where pictures depicting both (1) images for each sense of the word and (2) images for unrelated lemmas. After the round ends (Fig. 7.1c), players see (1) how accurate they were at destroying unrelated images and consistently choosing images associated with a single sense and (2) all of the pictures that they destroyed, both of which allows them to assess their performance and improve in future rounds. Our experiments show that Ka-boom! produces more accurate annotations than the current state-of-the-art game, Wordrobe (Venhuizen et al., 2013), and the highly-competitive MFS baseline. Because video games are an appealing past-time to many, this work raises the possibility of tapping a larger pool of video game players who may play games like Ka-boom! for free, significantly lowering the cost of annotation compared with that of crowdsourcing or that of trained annotators. In future work, we plan to scale up video game-based word sense annotation to both (1) increase the amount of sense-annotated data available for use and (2) explore the relationship between word senses and their visual representations.

7.2.2 Combining Distributional Semantics with Word Senses

Word meaning has typically been represented in two distinct forms: either with word senses (Navigli, 2009) or distributional representations (Turney and Pantel, 2010; Jurgens and Stevens, 2010b). The former divides meaning into discrete semantic units (i.e., senses), which are defined independently of how a word may be used. The latter representations builds upon the Distributional Hypothesis (Firth, 1957) to represent a word's meaning according to contextual features of its occurrences. While both repre-

sentations have proven highly useful to NLP, each comes with its own limitations, such as being unable to represent novel meanings within a fixed sense inventory (Lau et al., 2012), or using a single distributional representation that conflates multiple word meanings (Schütze, 1998). Accordingly, recent work has attempted to push each representation into a flexible middle ground, such as by allowing word usages to be represented with multiple senses (Erk et al., 2012; Jurgens, 2012), or by modifying a distributional representation based on the context of a particular usage (Erk and Padó, 2008; Dinu and Lapata, 2010). Nevertheless, models remain disconnected, forcing downstream applications to work either with senses or vectors.

In our ongoing work, we seek to bridge this divide by providing a flexible semantic representation that can operate as (1) a standard distributional vector, (2) a vector tuned to a word's meaning in context, (3) a distribution over word senses, or (4) even a single sense. Our hybrid approach leverages existing semantic networks over concepts and then enriches the network with distributional information. Our initial results show that this type of joint distributional- and sense-based representation can perform competitively on a wide variety of semantic tasks requiring either presentation and therefore offers the possibility of bridging the two representational divide.

7.2.3 Wikification and WSD

Wikipedia is a collaboratively-constructed encyclopedia where article texts commonly contain references that link to other articles within the encyclopedia. An ambiguous term such as "apple" is linked to the appropriate page, such as the company or fruit, based on its usage. Both Wikipedia and the process of linking a term to the appropriate page, also known as *wikification*, are close analogs of a word sense inventory and word sense disambiguation.

In our on-going work, we aim to link the tasks of wikification and WSD through alignments between Wikipedia pages and WordNet senses (Navigli and Ponzetto, 2010;

Niemann and Gurevych, 2011). Such an alignment enables gathering large amounts of manually sense-annotated data from the already-wikified links in Wikipedia, which can potentially alleviate the knowledge acquisition bottleneck for some words and also enrich the sense inventory with new senses from collaborative-constructed knowledge in Wikipedia. Conversely, WSID and WSD systems can be trained on Wikipedia as a sense inventory, providing a novel and potentially high-performance method of wikification.

7.2.4 Uncertainty and Underspecification in Sense Disambiguation

Our experiments with pseudowords and sense-annotated data have shown that linear kernel SVMs provide a consistent performance benefit for mapping induced senses into senses of a reference inventory. However, a potential downside to using this SVM setup is that the output always consists of a single sense label, which as we demonstrated in Chapter 5, is not always the most accurate representation of a particular usage's meaning. Two potential solutions could be explored in future work.

First, extensions to SVM have been proposed for multi-class classification (Knerr et al., 1990; Platt et al., 1999; Vural and Dy, 2004), where the SVM may report multiple labels. Furthermore, Wu et al. (2004) have proposed a method for estimating a probability distribution over all the labels. Such models could be applied to the mapping function to detect cases where the classifier identifies multiple senses for output.

Second, as we showed in Chapter 5, much of the need for having multiple WordNet senses label the same instance comes from WordNet's fine-grained distinctions in meaning. When we compared the multi-sense annotations with the annotation that would be made for the same instance with the more coarse-grained OntoNotes sense inventory, only a single OntoNotes sense annotation was needed in most cases. A second possibility for using single-sense classifiers like the linear kernel SVM is to use a two-pass approach to sense disambiguation where, first, the usage is disambiguated to

a single coarse-grained sense, and second, the sense label is refined to a fine-grained sense, if possible. The coarse-grain sense captures much of the ambiguity that would require multiple-fine grained senses.

APPENDIX A

Derivation of Fuzzy Normalized Mutual Information

This appendix chapter overviews the full derivation of mutual information described in Chapter 6 Section 6.2.1.2, which is used the evaluation of WSI systems in Section 6.3.5. This appendix provides additional details on the extension of Lancichinetti et al. (2009) and how our further extension for fuzzy clustering is incorporated. We begin by briefly reviewing the calculating of mutual information for hard (non-overlapping) clusterings and then describe the method of Lancichinetti et al. (2009). After, we describe our extension and how it may be computed.

Mutual information measures the dependence between two random variables and may be defined equivalently as

$$\begin{aligned} I(X:Y) = &\quad H(X,Y) - H(X|Y) - H(Y|X) \\ = &\quad H(X) - H(X|Y) \\ = &\quad H(Y) - H(Y|X) \end{aligned} \tag{A.1}$$

where $H(X)$ denotes the entropy of the random variable X that represents a partition, i.e., the sets of instances assigned to each sense, and $H(X|Y)$ is the conditional entropy.

In comparing hard clusterings, each clustering is treated as a discrete random variable that has been sampled N times where N is the number of items and where each outcome over the variable is an assignment to one of its clusters. The entropy of a particular assignment, X, may then be measured as follows,

$$\begin{aligned} H(X) &= \sum_{x_i \in X} p(x_i) \log_2(p(x_i)) \\ &= \sum_{x_i \in X} \frac{freq(x_i)}{N} \log_2 \left(\frac{freq(x_i)}{N} \right) \end{aligned}$$

where $p(x_i)$ denotes the probability of an item being assigned to cluster x_i and $freq(x_i)$ denotes the number of items that were assigned to cluster x_i. Similarly, for two partitions, X and Y, we may compute the conditional entropy as

$$\begin{aligned} H(X|Y) &= \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 p(x_i|y_j) \\ &= \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 \frac{p(x_i)}{p(x_i, y_j)} \\ &= \sum_{x_i \in X} \sum_{y_j \in Y} \frac{freq(x_i, y_j)}{N} \log_2 \left(\frac{\frac{freq(x_i)}{N}}{\frac{freq(x_i, y_j)}{N}} \right) \end{aligned} \tag{A.2}$$

The conditional entropy $H(Y|X)$ may be calculated in a similar manner, which enables calculating the mutual information (Eq. A.1).

To handle the case of mutual assignments, Lancichinetti et al. (2009) propose an extension to Eq. A.2 for calculating the normalized mutual information between soft clusterings, where items may be in more than one cluster. Rather than represent the entire clustering as a single variable, each cluster x_i is represented as a separate Bernoulli distribution; that is, the cluster assignments for each item is represented as a sample from the variable $\mathbf{X}_{1...k}$ denoting the clustering, where the ith entry of $\mathbf{X}$ is the Bernoulli distribution for cluster i. We denote the distribution of cluster i as X_i. Using the cluster labels for the N instances, we may easily compute the values of each component as $P(X_i = 1) = \frac{n_i}{N}$ where n_i is the number of items labeled with cluster i, and 1 denotes an instance is assigned to cluster i.

Given two soft clusterings, $\mathbf{X}$ and $\mathbf{Y}$, we can define the joint probability distribution between two of their individual clusters, X_i and Y_j, respectively, using their assignments. Let S_{X_i} denote the set of items in cluster X_i. The joint probabilities for the values of clusters X_i and Y_j may be calculated as

$$P(X_i = 1, Y_j = 1) = \frac{|S_{X_i} \cap S_{Y_j}|}{N} \tag{A.3}$$

$$P(X_i = 1, Y_j = 0) = \frac{|S_{X_i}| - |S_{X_i} \cap S_{Y_j}|}{N} \tag{A.4}$$

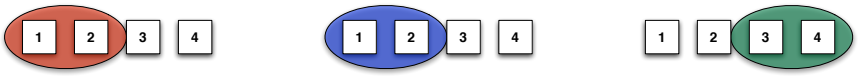
$$P(X_i = 0, Y_j = 1) = \frac{|S_{Y_j}| - |S_{X_i} \cap S_{Y_j}|}{N} \tag{A.5}$$

$$P(X_i = 0, Y_j = 0) = \frac{N - |S_{X_i} \cap S_{Y_j}|}{N} \tag{A.6}$$

Equations A.3–A.6 can then be used to calculate the conditional entropies of the two senses, i.e., for $H(X_i|Y_j)$, the amount of information that cluster Y_j gives us about the assignments of X_i.

$$\begin{aligned} H(X_i|Y_j) &= H(X_i, Y_j) - H(Y_j) \\ &= (h[P(X_i = 1, Y_j = 1)] + h[P(X_i = 1, Y_j = 0)] \\ &\quad + h[P(X_i = 0, Y_j = 1)] + h[P(X_i = 0, Y_j = 0)]) \\ &\quad - (h[P(Y_j = 0)] + h[P(Y_j = 1)]) \end{aligned} \tag{A.7}$$

In Equation A.7 and what follows, we adopt the shorthand of Lancichinetti et al. (2009) to denote $h(x)$ as $-x\log x$. Note that this value does not necessarily correspond the information that Y_j gives us about its items being in cluster X_i; the information from Y_j may also be high X_i is the complement. Figure A.1 illustrates the case where the conditional entropy is minimized both when the sense assignments are identical, shown in Figure A.1a and A.1b; however the conditional entropy is also minimized when the compared cluster is the complement, such as when comparing the clusters in Figure A.1a and A.1c. Therefore, Lancichinetti et al. (2009) impose the constraint that the conditional entropy is only defined if $h(p(X_i = 1, Y_j = 1)) + h(p(X_i = 0, Y_j = 0)) \geq h(p(X_i = 1, Y_j = 0) + h(p(X_i = 0, Y_j = 1))$, i.e., that the majority of the information is due to the degree of alignment rather than the degree of complement.



(a) X_i (b) Y_j (c) Y_k

Figure A.1: A cluster X_i in clustering $\mathbf{X}$ compared to two clusters, Y_j and Y_k in clustering $\mathbf{Y}$, where Y_j is identical to X_i and Y_k is the complement.

Under this constraint, the conditional entropy is minimized only when the two clusters are identically aligned. For example, in Figure A.1 the conditional entropy would be defined when comparing the clusters in Figures A.1a and A.1b, but not when comparing either to the cluster in Figure A.1c.

From Eq. A.7, we then define the conditional entropy from an individual cluster X_i to a clustering $\mathbf{Y}$ with n clusters as the minimum entropy from X_i to any of the clusters in $\mathbf{Y}$:

$$H(X_i, \mathbf{Y}) = \min_{j \in 1,2,\ldots,n} H(X_i|Y_j) \tag{A.8}$$

Should Eq. A.7 not be defined for all clusters in $\mathbf{Y}$ due to the matching constraint, $H(X_i, \mathbf{Y}) = H(X_i)$. Summing across the clusters in $\mathbf{X}$, the conditional entropy between $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$H(\mathbf{X}|\mathbf{Y}) = \sum_i H(X_i|\mathbf{Y}) \tag{A.9}$$

Finally, we may use the definition of conditional entropy between two soft clusterings (Eq. A.9) to compute the mutual information as defined in Eq. A.1. Lancichinetti et al. (2009) propose a normalization based on the variation of information (VI) criterion (Meilă, 2003):

$$\frac{1}{2}\left(\frac{\mathbf{X}|\mathbf{Y}}{\mathbf{X}} + \frac{\mathbf{Y}|\mathbf{X}}{\mathbf{Y}}\right)$$

However, as McDaid et al. (2011) note, this normalization unintentionally generates higher than expected values when both clusterings have clusters that contain the majority of the elements. To correct this issue, McDaid et al. (2011) propose an alternate method of normalizing suggested by Vinh et al. (2010) in their analysis of normaliz-

ing factors for mutual information. For two clusterings X, Y, the Normalized Mutual Information is defined as

$$\text{NMI} = \frac{I(\mathbf{X}, \mathbf{Y})}{\max(H(\mathbf{X}), H(\mathbf{Y}))} \tag{A.10}$$

In the fuzzy clustering setting, an instance's membership in a cluster is no longer binary (i.e., either a member or not), but rather a continuous value in in $[0, 1]$, reflecting the degree of applicability of that cluster's sense to the instance. Directly using continuous membership values when computing conditional entropy between clusters requires using differential entropy, which potentially produces negative values, creating problematic interpretations for comparing clusters. Therefore, instead of representing each cluster as a continuous random variable, the range of cluster membership values is discretized into disjoint subranges. In the case of sense annotation, this discretization is analogous to having annotators rate sense applicability with a Likert scale (i.e., a discrete value) rather than with a rational number within a fixed bound (e.g., $[0, 1]$).

Using the framework of Lancichinetti et al. (2009), discretization results in a cluster being represented as a categorical distribution with states denoting membership weights instead of a Bernoulli distribution whose $0/1$ states denote member/not-member states. Using the definition of X_i and Y_j as a categorical variables over discrete ratings, we may then estimate the entropy and joint entropy as follows.

$$H(X_i) = \sum_{i=1}^{n} p(w_i) \log_2 p(w_i) \tag{A.11}$$

where $p(w_i)$ is the probability of an instance being labeled with rating w_i. Similarly, we may define the joint entropy of two fuzzy clusters in separate clusterings as

$$H(X_k, Y_l) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(w_i, w_j) \log_2 p(w_i, w_j) \tag{A.12}$$

where $p(w_i, w_j)$ is the probability of an instance being labeled with rating w_i in cluster X_k and w_j in cluster Y_l, and m denotes the number of ratings bins for Y_l. Based on the limited range of fuzzy memberships in $[0, 1]$, we selected uniformly distributed

bins in $[0, 1]$ at 0.1 intervals when discretizing the applicability weights of the sense annotations. The conditional entropy may then be calculated as was defined in Equation A.7:

$$H(X_k|Y_l) = H(X_k, Y_l) - H(Y_l).$$

Together, Equations A.11 and A.12 maybe be used with the procedure of (Lancichinetti et al., 2009) for computing conditional entropies (Eqs. A.7 and A.9) to define $I(X, Y)$ for fuzzy clusters.

REFERENCES

Eneko Agirre 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*, pages 7–12. ACL, June.

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

Eneko Agirre, Olatz Ansa, David Martinez, and Eduard Hovy. 2001. Enriching WordNet Concepts with Topic Signatures. In *Proceedings of the NAACL Workshop on WordNet*.

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

Eneko Agirre, David Martínez, Oier ó de Lacalle, and Aitor Soroa. 2006b. Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 89–96. ACL.

Eneko Agirre, Oier López De Lacalle, Christine Fellbaum, Andrea Marchetti, Antonio Toral, and Piek Vossen. 2010. SemEval-2010 Task 17: All-words word sense disambiguation on specific domains. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. ACL.

Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. 2010. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764.

Ahmed N. Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalko. 2006. On similarity indices and correction for chance agreement. *Journal of Classification*, 23(2):301–313.

Héctor Martınez Alonso, Bolette Sandford Pedersen, and Núria Bel. 2013. Annotation of regular polysemy and underspecification. In *Procedings of the 51st Meeting of the Associatation for Computation Linguistics (ACL)*.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Iurii Derenikovich Apresjan. 1974. Regular polysemy. *Linguistics*, 142:5–32.

Guillaume Artignan, Mountaz Hascoët, and Mathieu Lafourcade. 2009. Multiscale visual analysis of lexical networks. In *Proceedings of the 13th International Conference on Information Visualisation*, pages 685–690. IEEE.

Javier Artiles, Enrique Amigó, and Julio Gonzalo. 2009. The role of named entities in web people search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–542. ACL.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Yoram Bachrach, Throre Graepel, Gjergji Kasneci, Michal Kosinski, and Jurgen Van-Gael. 2012. Crowd iq: aggregating opinions to boost performance. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 535–542. International Foundation for Autonomous Agents and Multiagent Systems.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop*, pages 563–566.

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

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Osman Baskaya, Enis Sert, Volkan Cirik, and Deniz Yuret. 2013. Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 300–306.

Michael S. Bernstein, Ggreg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology (UIST)*, pages 313–322. ACM.

Vikas Bhardwaj, Rebecca J. Passonneau, Ansaf Salleb-Aouissi, and Nancy Ide. 2010. Anveshan: a framework for analysis of multiple annotators' labeling behavior. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 47–55. ACL.

Chris Biemann and Valerie Nygaard. 2010. Crowdsourcing WordNet. In *Proceedings of the 5th Global WordNet Conference*.

Chris Biemann and Uwe Quasthoff. 2009. Networks generated from natural language text. *Dynamics on and of Complex Networks*, pages 167–185.

Chris Biemann. 2006. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80. ACL.

Chris Biemann. 2012. Turk bootstrap word sense inventory 2.0: A large-scale resource for lexical substitution. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 4038–4042.

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

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

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of International Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*, pages 31–40.

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

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

Samuel Brody, Roberto Navigli, and Mirella Lapata. 2006. Ensemble methods for unsupervised wsd. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 97–104. Association for Computational Linguistics.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources*, volume 2.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Paul Buitelaar. 2000. Reducing lexical semantic complexity with systematic polysemous classes and underspecification. In *Proeceedings of the NAACL-ANLP Workshop on Syntactic and Semantic Complexity in Natural Language Processing Systems*, pages 14–19. ACL.

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The salsa corpus: a german corpus resource for lexical semantics. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*.

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

Yee Seng. Chan, Hwee Tou Ng, and David Chiang. 2007. Word Sense Disambiguation Improves Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Wanxiang Che, Ting Liu, and Yongqiang Li. 2010. Improving semantic role labeling with word sense. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 246–249. ACL.

Timothy Chklovski and Yolanda Gil. 2005. Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In *Proceedings of the 3rd International Conference on Knowledge Capture*, pages 35–42. ACM.

Keith Chrzan and Michael Patterson. 2006. Testing for the optimal number of attributes in maxdiff questions. In *Proceedings of the Sawtooth Software Conference*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

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

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical Review E*, 70(6):66111.

Paul Cook and Graeme Hirst. 2011. Automatic identification of words with novel but infrequent senses. In *Proceedings of the 25th Pacific Asia Conference on Language Information and Computation (PACLIC 25)*, pages 265–274.

Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit players. 2010. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760.

Douglas Cutting, Jan Pedersen, David Karger, and John Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 318–329.

Ido Dagan and Alon Itai. 1994. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*, 20(4):563–596.

Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.

Antonio Di Marco and Roberto Navigli. 2012. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(4).

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172. ACL.

Beate Dorow and Dominic Widdows. 2003. Discovering corpus-specific word senses. In *Proceedings of 10th Conference of the European Chapter of the ACL*.

Philip Edmonds and Scott Cotton. 2001. Senseval-2: Overview. In *Proceedings of Senseval-2*, pages 1–5. ACL.

Wesam Elshamy, Doina Caragea, and William H. Hsu. 2010. KSU KDD: Word sense induction by clustering in topic space. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 367–370. ACL.

Katrin Erk and Diana McCarthy. 2009. Graded word sense assignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 440–449. ACL.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, HI. ACL.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2009. Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-AFNLP)*, pages 10–18. ACL.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2012. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554.

Stefan Evert. 2005. *The statistics of word cooccurrences*. Ph.D. thesis, Stuttgart University.

Christine Fellbaum, Joachim Grabowski, and Shari Landes. 1997. Analysis of a hand-tagging task. In *Proceedings of the ANLP Workshop on Tagging Text with Lexical Semantics*, pages 34–40.

Christine Fellbaum, Jaochim Grabowski, and Shari Landes. 1998. Performance and confidence in a semantic annotation task. *WordNet: An electronic lexical database*, pages 217–237.

Christine Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

J. R. Firth, 1957. *A synopsis of linguistic theory 1930-1955*. Oxford: Philological Society. Reprinted in F. R. Palmer (Ed.), (1968). Selected papers of J. R. Firth 1952-1959, London: Longman.

Radu Florian and David Yarowsky. 2002. Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 25–32. ACL.

Santo Fortunato. 2010. Community detection in graphs. *Physics Reports*, 486(3-5):75–174.

Marco Fossati, Claudio Giuliano, and Sara Tonelli. 2013. Outsourcing framenet to the crowd. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 742–747.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60.

Spandana Gella, Paul Cook, and Timothy Baldwin. 2014. One Sense per Tweeter... and Other Lexical Semantic Tales of Twitter. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword, ldc2003t05. *Linguistic Data Consortium*.

Silvana Hartmann, Iryna Gurevych, and Ubiquitous Knowledge Processing Lap. 2013. Framenet on the way to babel: Creating a bilingual framenet using wiktionary as interlingual connection. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.

H. Hassan, A. Hassan, and S. Noeman. 2006. Graph based semi-supervised approach for information extraction. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 9–16. ACL.

Barbora Hladká, Jiří Mírovskỳ, and Pavel Schlesinger. 2009. Play the language: Play coreference. In *Proceedings of the 2009 Joint Conference of the Annual Meeting for the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 209–212. Association for Computational Linguistics.

Jisup Hong and Collin F. Baker. 2011. How Good is the Crowd at "real" WSD? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL.

David Hope and Bill Keller. 2013a. MaxMax: A Graph-Based Soft Clustering Algorithm Applied to Word Sense Induction. In *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 368–381.

David Hope and Bill Keller. 2013b. UoS: A Graph-Based System for Graded Word Sense Induction. In *Proceedings of the Seventh International Workshop on Semantic Evaluations (SemEval)*, pages 689–694.

Eduard Hovy, 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)*, pages 57–60. ACL.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2(1):193–218.

Nancy Ide and Keith Suderman. 2004. The american national corpus first release. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 1681–1684.

Nancy Ide and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):2–40.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579.

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

David Jurgens and Ioannis Klapaftis. 2013. SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval), in the Second Joint Conference on Lexical and Computational Semantics*. ACL.

David Jurgens and Keith Stevens. 2010a. HERMIT: Using word ordering applied to the Sense Induction task of SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluations*. ACL.

David Jurgens and Keith Stevens. 2010b. The S-Space Package: An Open Source Package for Word Space Models. In *Proceedings of the ACL 2010 System Demonstrations*. ACL.

David Jurgens and Keith Stevens. 2011. Measuring the impact of sense similarity on word sense induction. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 113–123. ACL.

David Jurgens, Saif M. Mohammad, Peter D. Turney, and Keith J. Holyoak. 2012. SemEval-2012 Task 2: Measuring Degrees of Relational Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*. ACL.

David Jurgens. 2011. Word sense induction by community detection. In *Proceedings of Sixth ACL Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-6)*. ACL.

David Jurgens. 2012. An Evaluation of Graded Sense Disambiguation using Word Sense Induction. In *Proceedings of *SEM, the First Joint Conference on Lexical and Computational Semantics*. ACL.

David Jurgens. 2013. Embracing Ambiguity: A Comparison of Annotation Methodologies for Crowdsourcing Word Sense Labels. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of the Nineth International Conference on Language Resources and Evaluation (LREC)*.

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

Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34(1):15–48.

Adam Kilgarriff. 1998. Gold standard datasets for evaluating word sense disambiguation programs. *Computer Speech & Language*, 12(4):453–472.

Adam Kilgarriff. 1999. 95% replicability for manual word sense tagging. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 277–278. ACL.

Adam Kilgarriff. 2002. English lexical sample task description. In *Proceedings of ACL-SIGLEX SENSEVAL-2 Workshop*.

Aniket Kittur, Boris Smus, Suheel Khamkar, and Robert E. Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52. ACM.

Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. 1990. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer.

Ioannis Korkontzelos and Suresh Manandhar. 2010. UoY: Graphs of unambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 355–358. ACL.

Michal Kosinski, Yoram Bachrach, Gjergji Kasneci, Jurgen Van-Gael, and Thore Graepel. 2012. Crowd iq: Measuring the intelligence of crowdsourcing platforms. In *Proceedings of the ACM Conference on Web Science*. ACM.

Sotiris Kotsiantis and Dimitris Kanellopoulos. 2006. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*. Sage, Beverly Hills, CA.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA, second edition.

Ramesh Krishnamurthy and Diane Nicholls. 2000. Peeling an onion: The lexicographer's experience ofmanual sense-tagging. *Computers and the Humanities*, 34(1-2):85–97.

Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1003–1012. ACM.

Ravi Kumar and Sergei Vassilvitskii. 2010. Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 571–580. ACM.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015.

Shari Landes, Claudia Leacock, and Randee I Tengi. 1998. Building semantic concordances. *WordNet: An electronic lexical database*, 199(216):199–216.

Helen Langone, Benjamin R Haskell, and George A Miller. 2004. Annotating WordNet. In *Proceedings of the NAACL-HLT Workshop on Frontiers in Corpus Annotation*.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for computational Linguistics (EACL)*.

Jey Han Lau, Paul Cook, and Timothy Baldwin. 2013. unimelb: Topic modelling-based word sense induction. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 307–311.

Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. 2002. Discretization: An enabling technique. *Data mining and knowledge discovery*, 6(4):393–423.

J. J. Louviere. 1991. Best-Worst Scaling: A Model for the Largest Difference Judgments. Technical report, University of Alberta. Working Paper.

Ping Luo, Hui Xiong, Guoxing Zhan, Junjie Wu, and Zhongzhi. Shi. 2009. Information-theoretic distance measures for clustering validation: Generalization and normalization. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1249–1262.

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

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Yariv Maron, Michael Lamar, and Elie Bienenstock. 2010. Sphere embedding: An application to part-of-speech induction. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Tamara Martın-Wanton, Alexandra Balahur-Dobrescu, Andr'eas Montoyo-Guijarro, and Aurora Pons-Porrata. 2010. Word sense disambiguation in opinion mining: Pros and cons. *Special issue: Natural Language Processing and its Applications*, page 119.

Winter Mason and Siddarth Suri. 2012. Conducting behavioral research on amazons mechanical turk. *Behavior Research Methods*, 44(1):1–23.

Aaron F. McDaid, Derek Greene, and Neil Hurley. 2011. Normalized mutual information to evaluate overlapping community finding algorithms. arXiv:1110.2515.

Marina Meilă. 2003. Comparing clusterings by the variation of information. In *Proceedings of the Sixteenth Annual Conference ofn Computational Learning Theory (COLT)*. Springer.

Marina Meilă. 2007. Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Rada Mihalcea, Tim Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28. ACL.

George A. Miller, C. Leacock, Randee Tengi, and R.T. Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. ACL.

George A. Miller, Randee Tengi, and Shari Landes. 1998. Matching the tagging to the task. In Christine Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press.

Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):2.

G. Craig Murray and Rebecca Green. 2004. Lexical knowledge and human disagreement on a WSD task. *Computer Speech & Language*, 18(3):209–222.

Roberto Navigli and Giuseppe Crisafulli. 2010. Inducing word senses to improve web search result clustering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 116–126. ACL.

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

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 216–225. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2012a. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*.

Roberto Navigli and Simone Paolo Ponzetto. 2012b. Multilingual wsd with just a few lines of code: the babelnet api. In *Proceedings of the ACL 2012 System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Roberto Navigli and Daniele Vannella. 2013. Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application. In *Proceedings of SemEval-2013*.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of SemEval-2007*, pages 30–35.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10.

Mark E.J. Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Mark E.J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

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

Hwee Tou Ng, Daniel Chung Yong Lim, and Shou King Foo. 1999. A case study on inter-annotator agreement for word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources*.

Elisabeth Niemann and Iryna Gurevych. 2011. The people's web meets linguistic knowledge: Automatic sense alignment of wikipedia and wordnet. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS)*, pages 205–214, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cheng Niu, Wei Li, Rohini K. Srihari, and Huifeng Li. 2005. Word independent context pair classification model for word sense disambiguation. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 33–39. Association for Computational Linguistics.

Bryan Orme. 2009. Maxdiff analysis: Simple counting, individual-level logit, and hb.

Martha Palmer, Olga Babko-Malaya, and Hoa Trang Dang. 2004. Different sense granularities for different applications. In *Proceedings of the Second Workshop on Scalable Natural Language Understanding Systems*.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(02):137–163.

Rebecca Passonneau, Nizar Habash, and Owen Rambow. 2006. Inter-annotator agreement on a multilingual semantic annotation task. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1951–1956.

Rebecca J. Passonneau, Ansaf Salleb-Aouissi, and Nancy Ide. 2009. Making sense of word sense variation. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

Rebecca J. Passonneau, Ansaf Salleb-Aoussi, Vikas Bhardwaj, and Nancy Ide. 2010. Word sense annotation of polysemous words by multiple annotators. In *Proceedings of Seventh International Conference on Language Resources and Evaluation (LREC)*.

Rebecca J. Passonneau, Collin Baker, Christiane Fellbaum, and Nancy Ide. 2012a. The MASC word sense sentence corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*.

Rebecca J. Passonneau, Vikas Bhardwaj, Ansaf Salleb-Aouissi, and Nancy Ide. 2012b. Multiplicity and word sense: evaluating and learning from multiply labeled word sense annotations. *Language Resources and Evaluation*, 46(2):219–252.

Ted Pedersen and Varada Kolhatkar. 2009. Wordnet:: Senserelate:: Allwords: a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of NAACL, Demonstration Session*, pages 17–20. Association for Computational Linguistics.

Ted Pedersen. 2000. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 63–69. Association for Computational Linguistics.

Ted Pedersen. 2010. Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the SemEval 2010 Workshop : the 5th International Workshop on Semantic Evaluations*, pages 363–366, July.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Yves Peirsman, Kris Heylen, and Dirk Geeraerts. 2008. Size matters. Tight and loose context definitions in English word space models. In *ESSLLI Workshop on Distributional Lexical Semantics*, pages 34–41.

Mohammad Taher Pilehvar and Roberto Navigli. 2013. Paving the way to a large-scale pseudosense-annotated dataset. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 1100–1109.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).

John C. Platt, Nello Cristianini, and John Shawe-Taylor. 1999. Large margin dags for multiclass classification. In *nips*, volume 12, pages 547–553.

Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. 2013. Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation. *ACM Transactions on Interactive Intelligent Systems*, 3(1):3:1–3:44, April.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task 17: English lexical sample, SRL, and all-words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. ACL.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, pages 41–48. ACL.

James Pustejovsky. 1995. *The Generative Lexicon: A Theory of Computational Lexical Semantics*. Cambridge, MA: The MIT Press.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

John C. Raven. 1936. Mental tests used in genetic studies: The performances of related individuals in tests mainly educative and mainly reproductive. Master's thesis, University of London. Unpublished master's thesis.

Yael Ravin and Claudia Leacock. 2000. *Polysemy: Theoretical and computational approaches*. MIT Press.

Vassiliki Rentoumi, George Giannakopoulos, Vangelis Karkaletsis, and George A. Vouros. 2009. Sentiment analysis of figurative language using a word sense disambiguation approach. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, volume 9, pages 370–375.

Philip Resnik and David Yarowsky. 2000. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(3):113–133.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. ACL, June.

Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the Crowdworkers? Shifting Demographics in Mechanical Turk. In *Proceedings of the 28th of the International Conference on Human Factors in Computing Systems (CHI)*, pages 2863–2872. ACM.

Anna Rumshisky and Olga Batiukova. 2008. Polysemy in verbs: systematic relations between senses and their effect on annotation. In *Proceedings of the Workshop on Human Judgements in Computational Linguistics*, pages 33–41. ACL.

Anna Rumshisky, Nick Botchan, Sophie Kushkuley, and James Pustejovsky. 2012. Word sense inventories by non-experts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*.

Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. Framenet II: Extended Theory and Practice. `http://framenet.icsi.berkeley.edu/`.

Vasile Rus, Mihai Lintean, Art Graesser, and Danielle McNamara. 2009. Assessing student paraphrases using lexical semantics and word weighting. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK*.

Sawtooth Software. 2007. The maxdiff/web technical paper. Technical Report.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796.

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

Nitin Seemakurty, Jonathan Chu, Luis Von Ahn, and Anthony Tomasic. 2010. Word sense disambiguation via human computation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 60–63. ACM.

Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504.

Michael Shindler, Alex Wong, and Adam Meyerson. 2011. Fast and accurate k-means for large datasets. In *NIPS*.

Rion Snow, Brendan O'Connor, Dan Jurafsky, and Andrew Y. Ng. 2008. Cheap and fastbut is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263. ACL.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *3rd International Workshop on Semantic Evaluations (SensEval-3) at ACL-2004*, pages 41–43.

Anders Søgaard and Anders Johannsen. 2010. Robust semi-supervised and ensemble-based methods in word sense disambiguation. In *Advances in Natural Language Processing*, pages 401–405. Springer.

Stanley Smith Stevens. 1946. On the theory of scales of measurement. *Science*, 103(2684):677–680.

Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440.

J. Surowiecki. 2005. *The wisdom of crowds*. Anchor.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2000. Estimating the number of clusters in a dataset via the gap statistic. *Journal Royal Statistics Society B*, 63:411–423.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Akira Utsumi. 2010. Exploring the Relationship between Semantic Spaces and Semantic Relations. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, pages 257–262.

Tim Van de Cruys and Marianna Apidianaki. 2011. Latent Semantic Word Sense Induction and Disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1476–1485. ACL.

Stijn Van Dongen. 2000. A cluster algorithm for graphs. Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.

Cornelis Joost van Rijsbergen. 1979. *Information Retrieval*. Butterworths. *Second edition*.

Daniele Vannella, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli. 2014. Validating and extending semantic knowledge bases using video games with a purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*.

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

Jean Véronis. 1998. A study of polysemy judgments and inter-annotator agreement. In *Programme and advanced papers of the Senseval workshop*.

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

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

Nyugen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854.

Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 319–326.

Volkan Vural and Jennifer G Dy. 2004. A hierarchical method for multi-class support vector machines. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, page 105. ACM.

Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Warren Weaver, 1949. *Translation. Mimeographed*, page 12. John Wiley & Sons, July.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. 2000. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advance in Neural Information Processing Systems*.

Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. 2004. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(975-1005):4.

Xuchen Yao and Benjamin Van Durme. 2011. Nonparametric bayesian word sense induction. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*.

David Yarowsky. 1993. One sense per collocation. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 266–271, Plainsboro, N.J.

Deniz Yuret. 2012. FASTSUBS: An Efficient and Exact Procedure for Finding the Most Likely Lexical Substitutes Based on an N-Gram Language Model. *IEEE Signal Processing Letters*, 19(11):725–728, November.

Ying Zhao and George Karypis. 2005. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. ACL.

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

Dongqing Zhu and Ben Carterette. 2010. An analysis of assessor behavior in crowdsourced preference judgments. In *SIGIR Workshop on Crowdsourcing for Search Evaluation*, pages 21–26.