Journal of Artificial Intelligence Research 55 (2016) 1025–1058 Submitted 07/15; published 04/16

Semi-supervised Learning with Induced Word Senses for State of the Art Word Sense Disambiguation

Osman Başkaya OBASKAYA@KU.EDU.TR
Department of Computer Sciences and Engineering
Koç University
Istanbul, Turkey

David Jurgens JURGENS@STANFORD.EDU
Department of Computer Science
Stanford University
Stanford, CA, USA

Abstract

Word Sense Disambiguation (WSD) aims to determine the meaning of a word in context, and successful approaches are known to benefit many applications in Natural Language Processing. Although supervised learning has been shown to provide superior WSD performance, current sense-annotated corpora do not contain a sufficient number of instances per word type to train supervised systems for all words. While unsupervised techniques have been proposed to overcome this data sparsity problem, such techniques have not outperformed supervised methods. In this paper, we propose a new approach to building semi-supervised WSD systems that combines a small amount of sense-annotated data with information from Word Sense Induction, a fully-unsupervised technique that automatically learns the different senses of a word based on how it is used. In three experiments, we show how sense induction models may be effectively combined to ultimately produce high-performance semi-supervised WSD systems that exceed the performance of state-of-the-art supervised WSD techniques trained on the same sense-annotated data. We anticipate that our results and released software will also benefit evaluation practices for sense induction systems and those working in low-resource languages by demonstrating how to quickly produce accurate WSD systems with minimal annotation effort.

1. Introduction

Word Sense Disambiguation (WSD) identifies the particular meaning of a word in context, such as whether "bass" refers to a fish or an instrument. Correctly performing WSD grounds ambiguous natural language in a concrete semantic representation, which has numerous benefits to downstream applications, such as knowledge extraction (Navigli & Ponzetto, 2010; Hartmann, Gurevych, & Lap, 2013) and machine translation (Carpuat & Wu, 2007; Chan, Ng, & Chiang, 2007). Traditionally, supervised approaches to WSD have offered superior performance (Kilgarriff & Rosenzweig, 2000; Mihalcea, Chklovski, & Kilgarriff, 2004; Agirre & Soroa, 2007; Navigli, 2009). However, a major limitation for building high-performance supervised WSD systems has been the limited amount of sense-annotated corpora for training models on all word types, with the largest such corpora containing only hundreds of thousands of annotated tokens (Petrolito & Bond, 2014). As a result, unsupervised WSD techniques have been proposed to fill the need for high-coverage systems

(Yarowsky, 1995; Agirre et al., 2014; Moro et al., 2014). While these techniques are capable of disambiguating many word types, they have not surpassed the accuracy of supervised systems nor do they take advantage of what sense-annotated data is available.

An alternative approach to supervised WSD is to build semi-supervised approaches using Word Sense Induction (WSI), often referred to as Word Sense Induction and Disambiguation (WSID) models (Agirre et al., 2006). Word Sense Induction is a fully unsupervised technique that examines the contexts in which a word is used in order to learn (a) the word's different meanings, referred to as its *induced senses*, and (b) how to disambiguate a new usage of the word as an instance of one of those induced senses. The induced senses learned by a WSI method provide the key link for building a WSID system: by labeling usages with both induced senses and those from reference sense inventory such as WordNet (Fellbaum, 1998) or OntoNotes (Hovy et al., 2006), a mapping function can be learned to effectively transform an annotation using induced senses into an annotation using senses of the reference inventory. When a WSI model has been constructed from a large corpus of examples, its induced senses become associated with many contextually-disambiguating features. As a result, when the features associated with an induced sense are used in disambiguation, they are, through the proxy of sense mapping, being used as features for disambiguating between the reference senses –despite these contextual features potentially never being seen in the data annotated with references senses. Thus, when a close correspondence exists between induced and reference senses, a robust WSID system can be created that is ultimately able to disambiguate usages in contexts unlike those seen in data annotated with reference senses by virtue of the features associated with the induced senses. In contrast, the discriminatory capabilities of a supervised WSD system are limited to the features observed in the training data. Hence, the ability of a WSID system to leverage induced sense annotations can potentially remove the knowledge acquisition bottleneck of requiring significant amounts of sense-annotated data (Gale, Church, & Yarowsky, 1992).

Despite the potential of WSID, little analysis has been done into how to construct such models and how to maximize their performance. Instead, WSID systems have primarily been evaluated within SemEval tasks focusing on word sense induction (Agirre & Soroa, 2007; Manandhar, Klapaftis, Dligach, & Pradhan, 2010; Jurgens & Klapaftis, 2013). While WSID performance in such evaluations is promising, three important open questions remain. First, in current evaluations, WSID systems have all used the technique of Agirre et al. (2006) 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 of the results (Agirre & Martinez, 2000); these settings raise the question of how much of current systems' performances are attributable to the ease of disambiguating due to the test data's sense distribution. Third, despite the potential advantages of WSID for low-resource languages, no study has directly compared WSID and supervised WSD systems under equal conditions to test whether one setup should be preferred based on the amount of sense-annotated data available.

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 approximates all polysemous nouns in WordNet by using pseudowords to accurately model the difficulty of disambiguating those nouns. Here, a pseudoword is made of two or more monosemous lemmas, referred to as pseudosenses, each of which models a particular sense of a word. For example, the disemous noun *pic* has two WordNet senses: (1) a motion picture, and (2) a photograph. These two senses are represented by the monosemous nouns *movie* and *photo*, respectively. To simulate sense-annotated data, the occurrences of a pseudoword's pseudosenses are replaced by a unique token (e.g., replacing usages of *movie* and *photo* with a token denoting the pseudoword); then, in the analogous disambiguation task, a WSD system is shown an occurrence of the pseudoword and asked to decide which pseudosense was originally present. Crucially, because (1) these pseudowords approximate the real-world disambiguation difficulty and (2) pseudosense-annotated data can easily be created by sampling occurrences of the pseudosenses from a corpus, this resource enables performing a comprehensive evaluation of WSID on *arbitrarily-large* amounts of annotated data with direct generalizability to real-world WSD performance (Pilehvar & Navigli, 2014).

This paper offers the following four key 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– thereby providing statistically-robust results for all evaluated terms. Furthermore, 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 in order to measure its effect on performance. 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 in both pseudoword and real-world data. 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 acquisition bottleneck.

Our results offer two important practical implications. For researchers working with low-resource languages, our comparisons between WSID and supervised WSD demonstrate that only a relatively small amount of sense-annotated data is needed for state-of-the-art performance on WSD. Second, we demonstrate that the current approach to creating WSID systems artificially masks the true capabilities of the underlying WSI models, and thus future evaluations of WSID systems –such as those conducted in SemEval– should consider using the evaluation construction procedure described herein.

2. Word Sense Induction and Disambiguation 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 by Agirre et al. (2006). 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.

In constructing WSID systems, two key questions have not been examined: (1) the impact of the sense mapping function on WSID performance, and (2) whether multiple WSI models may be effectively combined. Answering these questions is essential to identifying the degree to which the performance of a WSID system is due to the capabilities of its underlying WSI model versus the mapping process used to create it. In what follows, we first describe the WSI models used in this paper to illustrate how they learn their senses. Then, we formalize the sense mapping function and define a range of possibilities for how it may be computed and propose how the function can be used to effectively combine WSI models into a WSID ensemble.

2.1 WSI Models

Multiple techniques have been proposed for how to effectively learn the different meanings of a word (Navigli, 2012), with many approaches using either (a) graph-based representations of a word's semantic relationships or (b) distributional approaches to identifying regularities in the word's contexts. Therefore, to increase the robustness of the results, four recent WSI methods were selected for our experiments. Models were balanced between those using lexical distributions and those using graphs: AI-KU (Baskaya, Sert, Cirik, & Yuret, 2013) and HDP (Lau, Cook, McCarthy, Newman, & Baldwin, 2012), which use token statistics to induce senses, and Chinese Whispers (Biemann, 2006) and SquaT (Di Marco & Navigli, 2012), which construct graphs to induce senses. Following prior evaluations (Manandhar et al., 2010; Jurgens, 2012; Jurgens & Klapaftis, 2013), for disambiguation, we allow these models to report multiple senses per context; in this setting, an induced sense denotes a prototypical meaning and the annotation represents how much the current usage resembles those meanings. Following, we summarize the models' induction and disambiguation procedures.

2.1.1 AI-KU

Baskaya et al. (2013) represent the context of each target word by using high probability lexical substitutes according to a statistical language model. 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, Lamar, & Bienenstock, 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.

2.1.2 HDP

Lau et al. (2012) propose a system that based on a Hierarchical Dirichlet Process (HDP) (Teh, Jordan, Beal, & Blei, 2006), a nonparametric extention of Latent Dirichlet allocation (Blei, Ng, & Jordan, 2003; Griffiths & Steyvers, 2004). HDP automatically infers both the number of topics and each topic's probability distribution for generating the tokens in the corpus. For sense induction, a HDP model is inferred from contexts of a target word, which produces a distribution over topics for each context. Each topic is treated as a distinct sense of the word. Given a new context of the word, the HDP model can be used to infer its topic distribution, thereby identifying which senses are present. For output, we report the full distribution over senses for each context, weighted by their probabilities.

2.1.3 Chinese Whispers (CW)

Biemann (2006) proposes inducing senses using the Chinese Whispers (CW), a nonparametric graph clustering algorithm. CW is a form of unsupervised label propagation where each vertex is initially assigned a unique label and then label propagation is run until either convergence or a fixed number of iterations have completed. When the graph is constructed of lexically-associated terms, vertices assigned to the same cluster typically form topically-related groups. For sense induction, a graph is constructed from words associated with a specific term (e.g., by computing statistical associations from a corpus) and after CW completes, each vertex cluster is considered as features of a distinct sense of the term. For CW, the graphs were constructed in three steps. First, the χ^2 association was computed for all words in the base corpus. Then, words associated with all of a word's pseudosenses are ranked by χ^2 and the 1000 words with the largest χ^2 are retained. For each of the retained neighbors, additional edges are added to each of its 1000 neighbors with the highest χ^2, excluding those edges to the pseudosenses. The sense features are then used to disambiguate new contexts by computing their overlap with content words of the context. For disambiguation, we report all senses containing at least one word in a context, weighted by the number of matching features.

2.1.4 SquaT

Navigli and Crisafulli (2010) construct a co-occurrence graph, which is then pruned to induce sense clusters. Term co-occurrences in the base corpus are scored using the Dice coefficient: For two terms w_1, w_2, $Dice(w_1, w_2) = \frac{2c(w_1,w_2)}{c(w_1)+c(w_2)}$ where $c(w)$ is the frequency of occurrence. Edges are added to the graph if their Dice coefficient is greater than a threshold δ. The graph construction begins with only co-occurrences of the target term and then proceeds to add edges of the newly-included neighbors. Their framework allows multiple pruning methods for induction; we adopt only the *Squares* pruning method, which was shown to perform best. Simply, edges are removed if the ratio of observed to potential squares (closed paths of length 4) in which an edge participates is below a threshold σ. Once pruned, the resulting disconnected components of the graph denote separate senses. For efficiency, we use only noun, verb, and adjective lemmas in the graphs. As the resulting graph produces sets of lemmas associated with each sense, sense disambiguation is performed in the same way as Chinese Whispers.

2.2 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 using a sense from a different inventory, with the induced senses essentially acting as features. Agirre et al. (2006) proposed the first mapping function based on matrix multiplication, which has been used by the 39 systems participating in WSID shared task evaluations (Agirre & Soroa, 2007; Manandhar et al., 2010; Jurgens & Klapaftis, 2013) and many subsequent papers on WSID (e.g., see Brody & Lapata, 2009; Klapaftis & Manandhar, 2010; Van de Cruys & Apidianaki, 2011; Lau et al., 2012; Wang, Bansal, Gimpel, Ziebart, & Yu, 2015). 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 $\mathbf{u}$ with non-zero values for the dimensions denoting the annotated senses. The product $\mathbf{u}M$ 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 is widely used by WSID systems, it comes with two limitations: (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 incorporated when an instance's induced labeling has multiple senses (Jurgens, 2012), with both due in part to the method's relative simplicity as a machine learning technique. Therefore, in constructing WSID systems, we evaluate six alternate 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. 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 the SVM classifier, whose instance weights were scaled into [0,1], and for Bernoulli naive 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).

2.3 An 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 & Yarowsky, 2002; Brody, Navigli, & Lapata, 2006; Søgaard & Johannsen, 2010), no work has pursued an analogous ensemble approach for WSID.[1] We propose a new heterogeneous ensemble WSID system built from the output of all four 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, as shown in Figure 1; the combined annotations are then used as features by the mapping function

1. We note that Stevens (2012) suggested using consensus clustering for sense induction as a way of creating an ensemble; however, no quantitative analysis was performed.

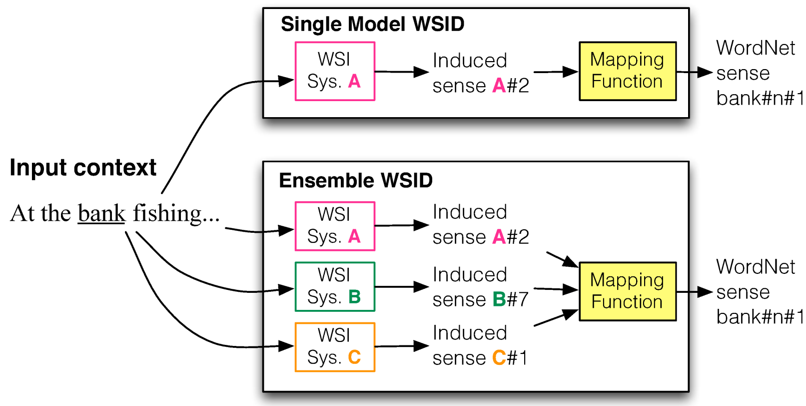


Figure 1: A comparison of the single-model and ensemble WSID systems, showing how the ensemble combines the output of multiple WSI models using a single mapping function.

to predict the sense. 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 senses in the reference sense inventory.

3. Experimental Design

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

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 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. For example, Schütze (1998) uses a pseudoword with the pseudosenses *banana* and *door*, which are semantically dissimilar; deciding between such a word's pseudosenses is akin to disambiguating between the sense of a homograph, which is known to be an easier disambiguation task (Ide & Véronis, 1998; Navigli et al., 2007). In contrast, most polysemous are not homographs and instead have senses that are semantic related in some way and therefore may appear in similar contexts (Apresjan, 1974; Rodd, Gaskell, & Marslen-Wilson, 2002; Palmer, Dang, & Fellbaum, 2007; Martínez Alonso et al., 2013). Thus, constructing pseudowords from arbitrarily-selected monosemous terms underestimates the difficulty of sense disambiguation and any results based on such pseudowords would not necessarily generalize.

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

Table 1: Examples of pseudowords for polysemous nouns in WordNet and the monosemous lemmas that comprise their pseudosenses

Pilehvar and Navigli (2013) propose a solution to the problem of appropriately choosing pseudosenses such that the disambiguation difficulty mirrors that of real-world data. A pseudoword dataset is created 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 inter-sense similarities of the corresponding polysemous word by mining WordNet's ontology to find monosemous words (pseudosenses) whose structural arrangement had close correspondence to that of the polysemous word's senses in the ontology. Because only the noun hierarchy of WordNet contains sufficient structure, their dataset was generated only for nouns.[2] While the inclusion of other parts of speech is ultimately desirable, nouns alone represent a significant challenge for WSD systems and multiple evaluations have focused entirely on disambiguating nouns (e.g., see Navigli, Jurgens, & Vannella, 2013). Table 1 shows example nouns and their corresponding pseudosenses used in the experiments.

The practical utility of these pseudowords was demonstrated by Pilehvar and Navigli (2014), who showed that the pseudowords' disambiguation difficulty accurately mirrored that of their corresponding polysemous words. Specifically, WSD systems were trained on the noun portion of the Senseval-3 dataset (Mihalcea et al., 2004) and a dataset made from those nouns' corresponding pseudowords. The resulting disambiguation performance on the pseudosense-annotated dataset was highly correlated with the performance on the Senseval-3 data. Their results indicate that by using their pseudowords, pseudosense-annotated datasets can be used to closely approximate real-world WordNet WSD performance, thereby avoiding the performance over-estimates caused by early methods of constructing pseudowords (Gaustad, 2001).

3.2 Data

All experiments were performed on a subset of data of Pilehvar and Navigli (2013). The original dataset includes pseudosenses that are likely to introduce noise in the results due to errors from part of speech tagging or when the word takes part in a named entity that was not present in WordNet. Therefore, to control for possible sources of noise, we exclude pseudosenses where (1) the lemma is also the plural form of another lemma, e.g., spirits, (2)

2. However, we note that in principle, such pseudowords could be constructed from the comparatively shallower verb hierarchy in WordNet and potentially for adjectives using the data of Tsvetkov et al. (2014).

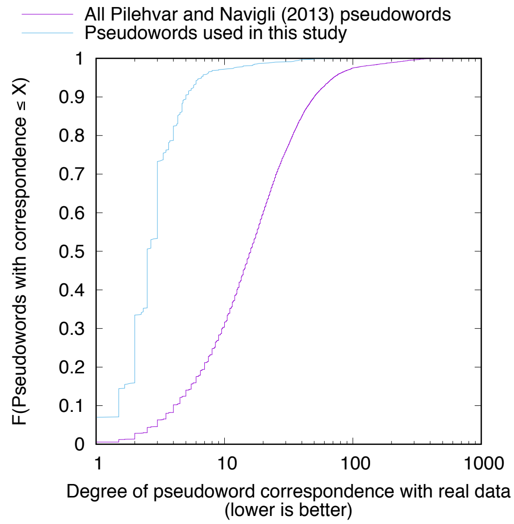


Figure 2: The cumulative distribution functions for the degree of correspondence between a word and its derived pseudoword, as specified in the dataset of Pilehvar and Navigli (2013). Lower values indicate a closer correspondence.

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 when part of speech tagging the corpus. The third criteria is necessary to ensure a sufficient number of instances are available for training and testing, which is discussed later in section 4.1.

The dataset provides a rating for each pseudoword indicating how closely its pseudosenses model the senses of the corresponding word in WordNet. For example, the pseudoword *doubles* shown in Table 1 has two pseudosenses, the monosemous words *tennis* and *badminton*, which closely model its two senses (1) "badminton played with two players," and (2) "tennis played with two players." Replacing one of these pseudosenses with the monosemous word *desk* would lower the resulting pseudoword's degree of correspondence since *desk* is not similar to either of the word's senses and thus, the disambiguation task would be potentially easier due to the dissimilarity of the contexts in which the pseudosenses appear.

The subset of their data used in our experiments was selected as follows. First, pseudowords are filtered according to the three aforementioned criteria. Second, the remaining pseudowords are ranked according to their degree of correspondence. Third, 920 senses were selected from this ranking to match the distribution of polysemy values in WordNet (e.g., having the same percentage of disemous lemmas). This third step was performed in order to ensure that the degrees of polysemy in our dataset are representative of the distribution in the full dataset of Pilehvar and Navigli (2013). Figure 2 shows the degree of correspondence between real words and (a) the pseudowords for used in our study and (b) those in the full dataset of Pilehvar and Navigli (2013), highlighting that the subset used in our experiments has significantly higher correspondence to real-world data than the full dataset and therefore is maximally representative of the expected real-world performance.

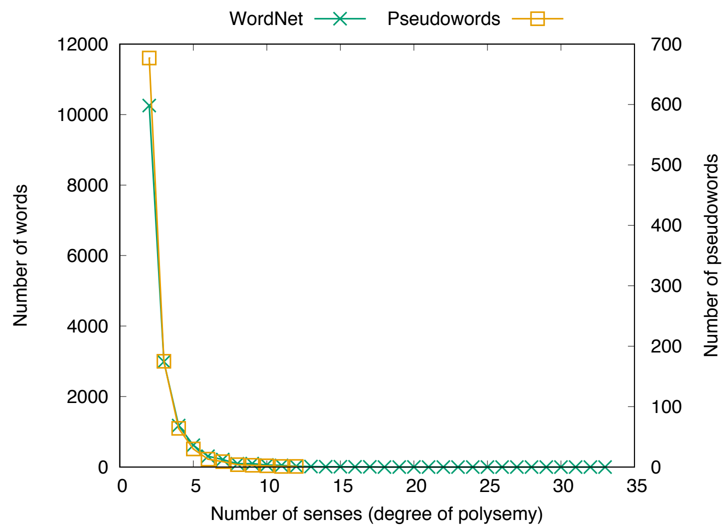


Figure 3: Distributions of the number of senses for polysemous nouns in WordNet and number of pseudosenses for our selected pseudowords, which were chosen to closely match the polysemy distribution in WordNet

Pseudowords in the final dataset had between two and twelve pseudosenses. Figure 3 shows the polysemy distribution of the number of senses for our selected pseudowords, compared with the polysemy distribution for all nouns in WordNet. Because the mapping functions described previously in Section 2.2 are parametric, five additional high-correspondence disemous pseudowords were also selected to use in parameter tuning, as this number of senses was the most common in our dataset and therefore most representative.

3.3 Sense Distributions

The frequency distribution of a word's senses is often peaked, with one or two senses occurring more frequently than the rest (Passonneau, Salleb-Aouissi, & Ide, 2009). The particular sense distribution of a word can greatly affect WSID performance, with artificially-inflated performance in settings where one sense occurs more frequently and all induced senses are mapped to that sense; in such cases, WSID performance is not generalizable to datasets where the sense distribution may vary. Controlling for the effect of the sense distribution in real-world test data requires a significant number of annotated instances from which to select, which is not currently possible with existing annotated corpora. However, when using pseudowords, the sense distribution may be precisely controlled by gathering the required number of usages of each pseudosense to match a desired distribution.

Precisely controlling the sense distribution allows us to measure WSID performance within two extremes. 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 noun with a corresponding pseudoword, we measure the frequencies of that noun's senses in SemCor, which determines the relative frequency of each of the pseudoword's pseudosenses. However, some words are still too infrequent in SemCor to accurately measure the expected frequency of their senses. 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 dataset as having a SemCor sense distribution.

The SemCor distribution measures the difficulty of WSID in the expected setting where some senses are more likely to occur than others. However, the presence of a majority class can potentially mask important underlying performance differences between systems; because one sense is more likely, a model's performance is not necessarily representative of its ability to distinguish between all senses of a word. Therefore, for the second distribution, all of a word's senses appear with uniform probability. When both training and test data have a uniform sense distribution, WSD systems cannot use the often-effective strategy of always choosing the most-frequent sense seen in the training data. Furthermore, the Uniform-distributed data allows measuring the ability of the sense mapping function to find a correspondence between induced and reference senses when induced senses have equivalent amounts of data. While the uniform sense distribution is not representative of real-world data, a comparison between a model's performances on SemCor- and Uniform-distributed datasets provides critical insight into its disambiguation capabilities and its expected generalizability to new data with arbitrary perturbations in the underlying sense distribution. For example, if a model performs well on SemCor-distributed data but not on Uniform-distributed data, this result suggests that the model is only effective at identifying the most-frequent sense; in contrast, models that perform well on both SemCor- and Uniform-distributed data would be expected to maintain its accuracy on data with other sense distributions based on its ability to discriminate between the senses when no one sense is more frequent.

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. (2006) mapping function, we assess whether any of the six alternatives described in Section 2.2 can consistently improve WSID performance. Second, we assess whether the proposed sense mapping functions can effectively fuse the induced sense annotations of multiple WSI models to produce an accurate ensemble model.

4.1 Experimental Setup

In what follows, we detail the parameters and training of the WSI systems and how the training and test data was constructed.

4.1.1 WSID Systems

WSI models were trained on the same base corpus, ukWaC (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009), though we emphasize 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 as reported by Baskaya et al. (2013). Using the same setup for SemEval 2013 WSI task, AI-KU calculates the lexical substitutes by using SRILM (Stolcke, 2002) with the ukWaC

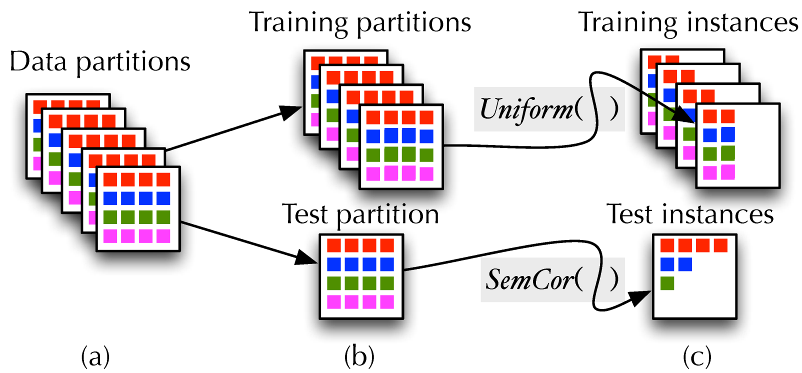


Figure 4: A schematic of the cross validation. Data partitions initially contain an equal number of instances per pseudosense (a), shown as different colored boxes. For each fold of validation, four partitions are used for training and one for test (b). The instances from each partition are then sampled according to a distribution (shown in italics), which produces the training and test datasets (c).

(Ferraresi, Zanchetta, Baroni, & Bernardini, 2008) as a corpus to construct its 4-gram language model. 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 by Lau et al. (2012) and Lau, Cook, and Baldwin (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 all pseudowords. The Chinese Whispers model is nonparametric, so no parameter choices are needed.

In total, twenty eight WSID configurations were built from each combination of the seven sense mapping functions (Sec. 2.2) and four WSI models (Sec. 2.1). Additionally, 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.

4.1.2 Cross-Validation Evaluation

Systems were evaluated using five-fold cross validation, with modifications to ensure that folds were of comparable sizes between distribution types and that no training data leaked into test data when changing the sense distribution of the test data. Initially, the corpus of all instances of a pseudoword is divided into five partitions, where each partition contains 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 4 visualizes this process. For evaluation, four filtered partitions form the training data and one partition is used as test data. Importantly, this setup ensures that its instances remain 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 on the same training data from identical folds, ensuring the separation between test and training data.

The reported experiments use the same sense distributions in both training and testing (either SemCor or Uniform). However, our evaluation setup is sufficiently general to support using arbitrary distributions, including different distributions between training and testing data, as shown in the example in Figure 4; results using additional combinations of distributions are reported in the Supplementary Material.

4.1.3 Evaluation Data

All data for the partitions was drawn from the Gigaword corpus (Graff, Kong, Chen, & Maeda, 2003). Instances of pseudosenses were filtered to ensure the correct part of speech and to remove all occurrences where the pseudosense was part of a named entity. Ultimately each partition in the test 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 comparatively fewer instances in the fixed-size setting and in which case, the WSID accuracy would not be significantly affected by the model's ability to identify such senses. 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, evenly distributed between senses.

Two baselines are used with the test 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 has outperformed many WSID models in previous studies (McCarthy et al., 2004; Kilgarriff, 2004; Navigli, 2009). Note that in the Uniform sense distribution, the MFS and Random baselines are equivalent.

4.1.4 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. When a system labels all instances, precision and recall are equivalent. Because the number of instances per term scales with the number of senses, precision and recall can be considered microaverages of WSD performance across words.

4.1.5 Parameter Tuning

Five disemous words were used to tune each parametric mapping function. Using a grid search over parameter values, each WSID configuration 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's mapping function.

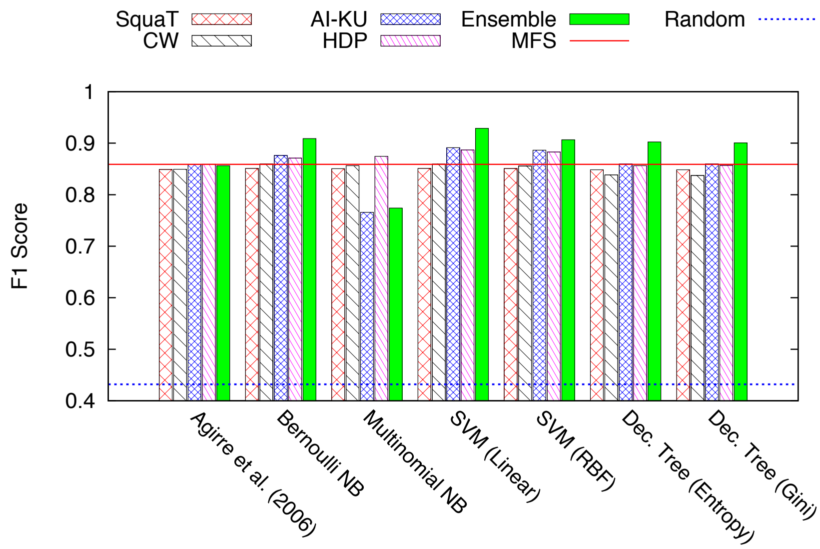


Figure 5: Average performance of all WSID systems when training and testing data follow the SemCor sense distribution

4.2 Results and Discussion

The results of WSID systems using a single WSI model demonstrate that high sense disambiguation performance is possible when using a suitable sense mapping function and that multiple WSI models may be effectively combined into an ensemble.

4.2.1 Single-Model Results

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 5.[3] For nearly all systems, the Agirre et al. (2006) method mapped all induced senses to the most-frequent sense seen in the SemCor-distributed training data, ignoring any sense distinctions recognized by the WSI model. While the Agirre et al. method does produce WSID systems that outperform those using the Multinomial Naïve Bayes, the performance says little about the discriminative capabilities of the WSID systems and effectively prevents meaningful comparison, hindering the testing and development of new WSID systems.

In contrast to the performance when using the Agirre et al. mapping function, the SVM and Bayesian functions both produced two WSID systems that outperformed the MFS baseline. The best performance when using a single WSI model comes from SVM with a linear kernel, which provides slightly higher performance than a RBF kernel. Indeed, a WSID system using the AI-KU model and a linear kernel SVM mapping function has a

3. For all WSID systems, tuning the parameters of the mapping function provided little to no performance improvement in either sense distribution. We therefore omit the tuned results here for brevity, but report these scores in the Supplementary Materials.

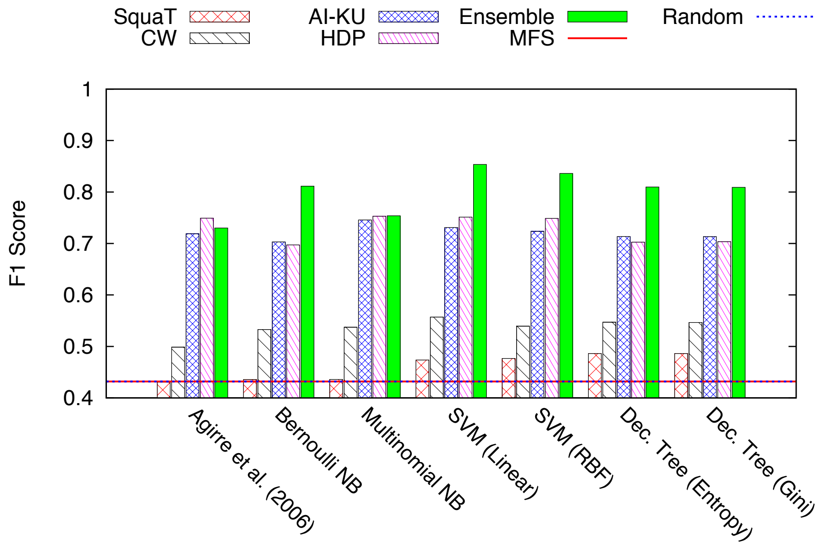


Figure 6: Average performance of all WSID systems when training and testing data follow the Uniform sense distribution

3.8% increase in F1 score over the MFS baseline, which is statistically significant at $p < 10^{-6}$ using McNemar's test of significance.

The impact of the choice in sense mapping function on WSID performance is even more evident in the Uniform-distribution dataset. The results, shown in Figure 6, reveal significant differences in the discriminatory capabilities of WSID systems. WSID systems using Agirre et al. mapping function perform well on average, indicating that the function is capable of learning an effective correspondence between senses when no single sense dominates in frequency. Nevertheless, all WSI models enjoy consistently-higher performance when using a SVM mapping function, which all have a statistically significant improvement at $p < 10^{-6}$. Furthermore, even the worst-performing model, SquaT, is still able to more than double the performance of the MFS baseline when using SVM or Decision Tree mapping functions.

The single-model results on the Uniform distribution also provide insight into how models would be expected to perform on new datasets where sense distributions differ from that in the SemCor-distributed data. Systems' performances were relatively close when tested on the SemCor dataset and differed by at most 0.04 F1 with the linear-kernel SVM; in contrast, systems differed by more than 0.278 F1 with the Uniform-distributed data, indicating significant differences in the WSI models' abilities to find meaningful sense distinctions. Furthermore, a clear difference is seen between distributional and graph-based WSI approaches, suggesting that distributional techniques may be more robust to potential changes in a corpus's sense distribution.

The overall ranking between individual-model WSID systems is consistent across the different mapping functions. However, some rank oscillation does appear between the HDP and AI-KU models in the Uniform distribution setting and between the CW and SquaT models in the SemCor distribution setting. In both cases, the SVM-based mapping functions provide the highest average performance across systems and produce identical rankings. As such, we view the ranking differences with other mapping functions to be an artifact of the mapping function itself, rather than due to actual performance differences between the systems.

4.2.2 Ensemble-Model Results

In nearly all WSID configurations, the ensemble WSID system obtains substantial performance gains over both the MFS baseline and the best WSID system built from a single WSI model. For SemCor-distributed data (Fig. 5), the ensemble with a linear kernel SVM produces the highest performance of all WSID configurations, achieving a 9.4% increase in F1 over the MFS and 4.2% increase over the next-closest system (AI-KU). Furthermore, except when using the Agirre et al. (2006) and Multinomial Naïve Bayes mapping functions, the 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 6, with a 13.6% increase in F1 over the next-closest system.

The results for both distributions indicate that using a linear kernel SVM for sense mapping provides consistently superior WSID performance that is robust to variations in the choice of WSI model. Furthermore, if the annotations with induced senses contain complementary sources of information, as in the case of the ensemble sense labeling, the SVM mapping function is able to produce better quality sense annotations.

The success of the ensemble with using all mapping functions *other than* the method of Agirre et al. (2006) highlights a potential obstacle in the research community: While prior attempts at building ensemble WSID methods may have been considered, any performance benefit would not have been observed due to the current community-wide practice using the method of Agirre et al. Further, our work raises the possibility that new mapping functions could be developed to more-effectively combined induced senses.

4.2.3 Quantifying the Impact of Polysemy on Disambiguation Performance

Given the high performance of using a linear-kernel SVM as a mapping function, we performed a follow-up analysis to measure its performance effect relative to the number of senses per word. This analysis separates the improvement from the relative difficulty of disambiguation and provides a more-complete picture of WSID performance. Performances for pseudowords with six or more senses were combined due to the words' relative infrequency. Figures 7 and 8 show the performances per term for SemCor and Uniform sense distributions, respectively, using a box and whisker plot. Whiskers denote the maximum and minimum F1 for any pseudoword, boxes denote the first and third 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 7 and 8, the ensemble WSID system offers superior performance across all levels of polysemy. For example, although one sense of disemous words occurs in the vast majority of instances in the SemCor data, the ensemble WSID performance is still able to surpass this MFS baseline for nearly all words (as shown in the left-most

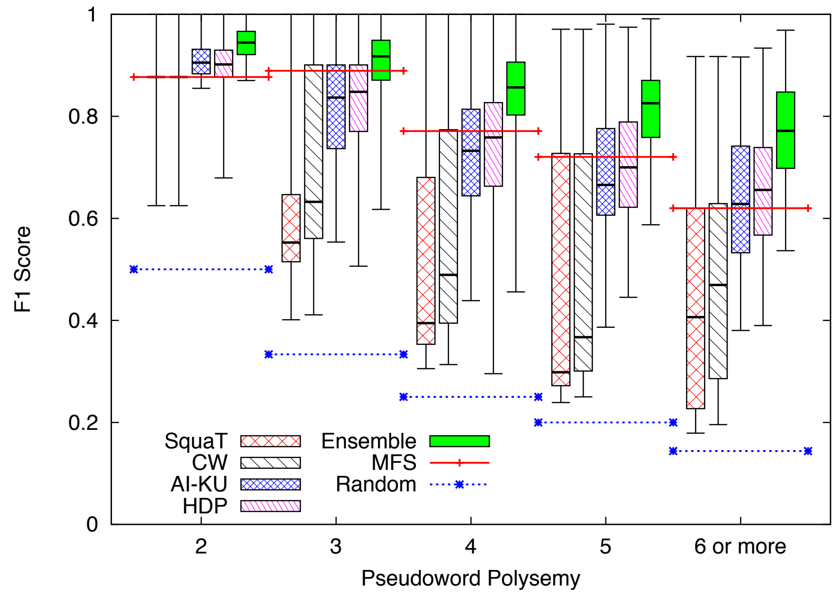


Figure 7: Performance of WSID systems using a linear-kernel SVM for different polysemy on SemCor-distributed training and testing data

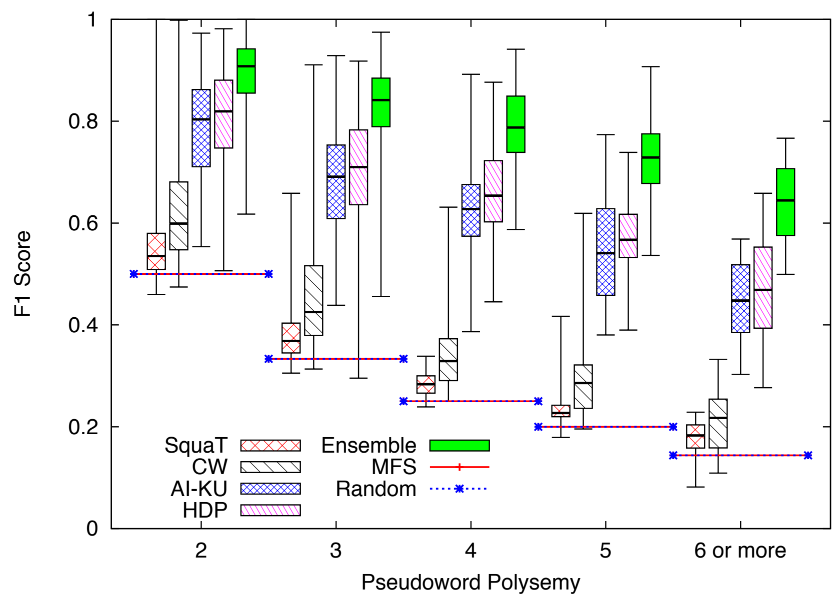


Figure 8: Performance of WSID systems using a linear-kernel SVM for different polysemy on Uniform-distributed training and testing data

cluster of boxes in Figure 7). The results of both settings indicate that the ensemble WSID model would offer superior performance on new data with arbitrary sense distributions. Indeed, in the current datasets, most words had either two or three senses (87%), for which the ensemble WSID model sees its smallest improvement over the MFS. If evaluated on a corpus containing words that have more senses, the overall performance improvement of the WSID ensemble over the MFS baseline would be even higher than reported in the main results of Experiment 1 (Figure 5).

5. Experiment 2: Comparing WSID and Supervised WSD

When sufficient sense-annotated data is available, supervised machine learning has typically been shown to produce the best-performing WSD systems. However, the results of Experiment 1 indicate high WSD performance is also possible using a semi-supervised WSID model. As both approaches require some amount of sense-annotated data, this raises the question of under what circumstances should one approach be expected to outperform the other. Therefore, in Experiment 2, we perform a direct comparison between semi-supervised WSID systems and the current state of the art for supervised WSD, using identical training data. The results of this experiment have direct implications for sense annotation efforts in deciding how much data is necessary for high performance.

5.1 Experimental Setup

In what follows, we describe the configuration of the supervised WSD system used for comparison and how training data was created.

5.1.1 Supervised WSD

For comparison, we use It-Makes-Sense (IMS) (Zhong & Ng, 2010), a state-of-the-art supervised WSD algorithm. To disambiguate a usage in a sentence-length context, IMS extracts features consisting of the neighboring lemmas and their POS along with neighboring collocation pairs. IMS uses linear-kernel SVM on these feature vectors for predicting the sense. In our experiments, IMS was trained using the default algorithmic parameter values specified in its publicly-available implementation.

The experiments are intentionally not measuring the disambiguation ability of the fully-trained IMS system provided by its authors;[4] rather, the experiments are intended to directly compare the results using the current state-of-the-art supervised WSD algorithm. While it would be possible to retrain WSID systems on the training data used by the authors' fully-trained model, the annotated corpora used in the original experiments are not readily available nor are the sense distributions of those corpora controlled for, making the conclusions of such an experiment difficult to generalize.

5.1.2 Training and Test Data

For Experiment 2, multiple datasets are created with increasing amounts of training data in order to measures the ability of WSID and supervised WSD in each condition. The datasets are generated similarly to how instances were allocated for sense distributions in Experiment 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 based on their relative frequency. Because of the different number

4. `http://www.comp.nus.edu.sg/~nlp/software.html`

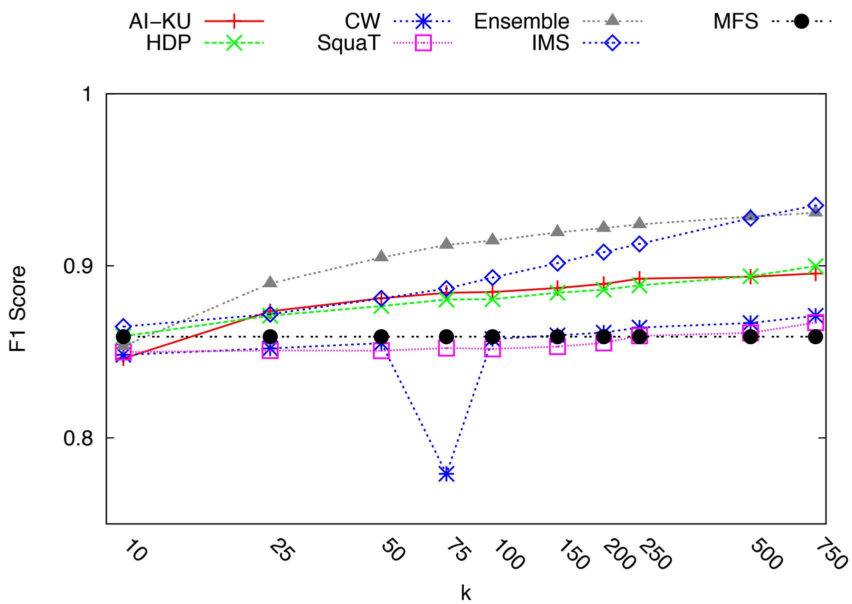


Figure 9: Performance of IMS and WSID systems on SemCor-distributed data

of instances per word in the SemCor-distributed training data, the Uniform-distributed data is created in such a way to account for the difference: Given a specific k and word with n total instances for its m senses, the corresponding Uniform-distributed training data is constructed by including $\frac{n}{m}$ instances for each of the pseudoword's m senses. For notational clarity, we use $\hat{k}$ to denote the equivalently-sized Uniform-distributed dataset whose corresponding SemCor-distributed training data has k instances.

Training and test data were generated from the same Gigaword data used in previous experiments, using five folds for 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 $\hat{k}$ datasets created from four partitions and then tested on the fifth, *full* partition. Because the test set is identical to that of Experiment 1 and all instances are used in testing, the resulting performances for each k and $\hat{k}$ are directly comparable to the results of Experiment 1.

5.1.3 WSID Systems

WSID systems were constructed using the same procedure used in previous experiments (cf. Sec. 4.1). For simplicity, we report only WSID systems using a linear kernel SVM, as these provided the highest performance. Full results for all other configuration are available in the Supplementary Material.

5.2 Results and Discussion

The results reveal that WSID systems can offer superior performance over IMS when few annotated instances are available. Figures 9 and 10 show the resulting F1 scores for

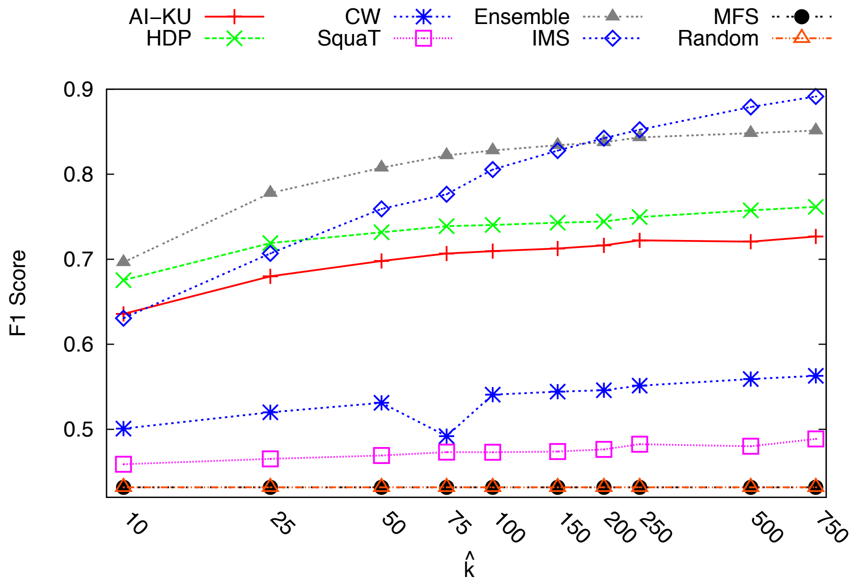


Figure 10: Performance of IMS and WSID systems on Uniform-distributed data

SemCor- and Uniform-distributed data, with x-axis drawn at log scale. The random baseline (F1=0.432) is omitted from Figure 9 for better visual contrast.

In SemCor-distributed data, IMS outperforms all single-model WSID systems for nearly all values of k, though the AI-KU and HDP models are closely competitive differing by less than 1% F1 for $k \leq 75$. In contrast to the single-model WSID systems, the ensemble WSID system outperforms IMS starting at k=25 until just after k=500. All ensemble performance differences $25 \geq k \geq 250$ are statistically significant at $p < 10^{-6}$ and the IMS and WSID performances at k=500 are statistically equivalent. Given that the publicly-distributed IMS system was trained on an average of 35 instances per word type, our results suggest that training an ensemble WSID model on the same data would provide superior performance.

When very few training instances are available, for most cases words, the IMS algorithm was able to correctly learn the back-off strategy of always selecting the most frequent sense from the training data, thereby ensuring it performs at least as well as the MFS baseline. In contrast, the mapping function for WSID models is slightly noisier and does not learn an accurate mapping from induced senses to reference senses, resulting in performance just below the MFS baseline.

A similar trend is seen when testing on Uniform-distributed data (Figure 10), though the ensemble WSID system outperforms IMS at $\hat{k} = 10$ until $\hat{k}$=200, at which point they are statistically equivalent, and the HDP model initially outperforms IMS as well until just after $\hat{k} = 25$. The results of the Uniform-distributed setting indicate that WSID models can provide accurate discriminatory techniques.

Together these results suggest that 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 also 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 from annotated data alone.

Last, we note that increasing the amount of training data consistently improves the performance of IMS, while providing decreasing benefit to WSID models. This contrast highlights the difference in how both systems learn. Training a WSID model on additional sense-annotated data cannot directly improve disambiguation performance as the contextual features used for disambiguation are fixed by the underlying WSI model, which is independent of the training data. In contrast, providing the same data to a supervised WSD system may enable it to learn new features for disambiguation. Nevertheless, the performance of WSID does depend on having sufficient sense-annotated data to train a correct mapping function, as shown in the large performance improvements between k=10 and k=25 shown in Figures 9 and 10 where the increase in training data provides a substantial improvement in sense mapping.

6. Experiment 3: Evaluation with SemEval Systems

Experiments 1 and 2 demonstrated that WSID models are capable of accurate WSD and that when individual WSI models are combined into an ensemble, the resulting system is capable of outperforming fully-supervised WSD. However, both experiments were performed in controlled conditions on pseudoword data. Therefore, in Experiment 3, we test whether the observed performance improvements carry over to real-world data. In three tests using over thirty WSI models and two sense inventories, we evaluate the impact of our new mapping function and ensemble construction in extensions of prior WSID evaluations.

6.1 Experimental Setup

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

Two significant differences exist in the tasks' setup compared with our earlier experiments. The 2007 and 2010 tasks use OntoNotes senses (Hovy et al., 2006), which are known to be more coarse-grained than the WordNet senses that the pseudowords models. Second, the 2013 task also focuses on instances where multiple meanings may be evident (e.g., due to ambiguity or syllepsis) and therefore includes some gold standard data where instances have multiple sense labels. Because our experimental setup has focused only on instances with one sense interpretation, we adopt the single-sense evaluation described by Jurgens and Klapaftis (2013) using the subset of 4122 instances in the task data that have a single sense annotation.

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 four models used in our earlier experiments in order to test the benefits of the proposed WSID configuration in different settings and to quantify its generalizability. However, we note that the two highest-performing

			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	0.640	0.640	**0.657**

Table 2: A comparison of the best-performing system in each SemEval WSID task and our proposed ensemble method.

WSI systems in the prior experiments, AI-KU and HDP, also participated in the 2013 task, so the ensemble results of that task are expected to be similar. 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.

6.2 Results

The results of both ensemble and single WSI-model systems, described next, demonstrate the benefits of using the new WSID construction procedure.

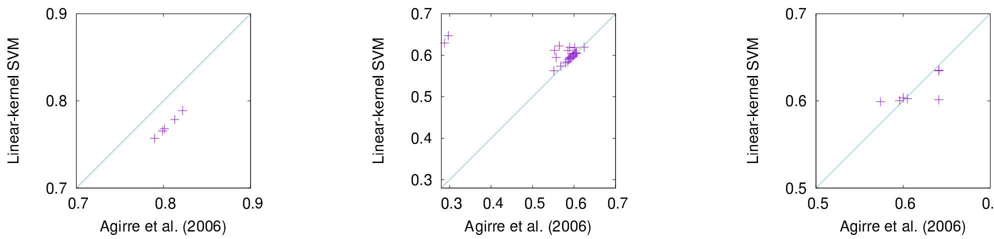
6.2.1 Ensemble Results

For all three tasks, the ensemble WSID configuration shows performance improvements over both the best-performing system and MFS. Table 2 shows the results for all three tasks, including the scores of best-performing system in each task originally and the task's MFS score. Improvements over MFS were all significant at $p < 10^{-6}$ using McNemar's test of significance. Similarly, the improvements over the best systems in each task are significant at $p < 10^{-6}$ for all ensemble configurations, with the exception of the ensemble for SemEval-2007, which is significant at $p < 10^{-4}$. Furthermore, we note that the improvement by the ensemble in each task is larger than the difference between the task's best and second-best systems, indicating a substantial increase in performance. 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.

6.2.2 Single-Model Results

Single-system WSID models varied in whether the use of an SVM mapping function improved performance, as shown in Figure 11.[5] For SemEval-2007, systems obtained a lower F1 when using the SVM, with an average decrease of 0.032 but no change in the overall system ranking. In contrast, systems performed better for the 2010 and 2013 tasks when using an SVM, with average F1 increases of 0.043 and 0.004. Although these mixed trends initially seem contradictory to the prior experiments' results showing a consistent benefit from the SVM, the performance differences are partly due to differences in the task setup

5. Full score details are available in the Supplementary Material.



(a) SemEval-2007 (b) SemEval-2010 (c) SemEval-2013

Figure 11: Comparisons of the F1 scores for each system in the SemEval tasks when using the linear-kernel SVM mapping function (y-axis) versus that of Agirre et al. (2006) (x-axis). Points above the diagonal indicate an improvement when using the SVM function.

where most WSI systems are designed to label each usage with only a single induced sense. In contrast, the systems in our prior experiments reported multiple induced senses per instance, weighted by how applicable the sense was to the instance. The multiple senses provide a richer feature set for training and enable recognizing cases where lower-weighted induced senses provide information on the correct sense annotation. When the WSI system reports only a single sense, WSID system performance has an the upper-bound based on the reference sense with the highest conditional probability, given an induced sense.

Even when a single induced sense is reported, using a SVM mapping function can still significantly impact resulting performance, as shown in the 2010 task. Here, multiple systems in the lowest ranked achieved significant improvements in F1 with some seeing over 0.30 absolute increases. The performances differences also highlight a unique feature of the 2010 task; Pedersen (2010) submitted four systems that generated random sense assignments, which were ranked as high as 18th out of the 26 systems. The SVM-based ranking correctly assigns the four random-answer submissions to the lowest four ranks. Indeed, the overall system ranking for the task changes dramatically from the originally-reported ranking (Spearman's ρ=0.14). Our results (Fig. 11b) suggest that while all systems are performing above random chance, the systems actually differ little in their abilities on the task and that some previously low-ranked systems actually offer superior performance. Thus, while the overall performance on the task is not as high, the SVM-based moel reveals the true discriminatory capabilities of the WSI systems, which were partially masked by having many induced senses mapped to the most frequent reference sense, artificially increasing performance.

7. Related Work

The present study touches upon three bodies of prior work on word sense induction and its relationship with WSD, semi-supervised WSD, and work on pseudowords.

7.1 Word Sense Induction and Disambiguation

Purandare and Pedersen (2004) and Niu, Li, Srihari, and Li (2005a) 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 2, Agirre et al. (2006) 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, suggesting that high WSID performance is possible. Last, Jurgens (2012) notes the potential for having WSI models annotate items with multiple senses and proposes a modification to the mapping function of Agirre et al. 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.

7.2 Pseudowords

Since first being proposed for word sense disambiguation (Gale et al., 1992; Schütze, 1992), pseudowords have been incorporated into evaluations for multiple tasks, with specific recommendations for how to improve their construction for tasks such as modeling selectional preferences (Chambers & Jurafsky, 2010), modeling word co-occurrence (Dagan, Lee, & Pereira, 1999), machine translation (Duan, Zhang, & Li, 2010), tasks in Information Retrieval (Stokoe, 2005) and even improving word embeddings (Liu, Liu, Chua, & Sun, 2015). Indeed for WSD, new techniques have been proposed for adapting pseudowords to other languages such as Chinese (Lu, Ting, & Sheng, 2004; Lu, Wang, Yao, Liu, & Li, 2006) and for creating pseudowords that more accurately model the difficulty of WSD (Nakov & Hearst, 2003; Otrusina & Smrz, 2010; Pilehvar & Navigli, 2013), though only the approach of Pilehvar and Navigli (2013) –which is used here– was shown to closely correlate with real-world performance.

Most related to our study are those work analyzing word 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. To test the discriminatory ability of WSI models, Jurgens and Stevens (2011) create a set of disemous pseudowords where the pseudosenses have varying degrees of similarity. To represent the full range of pseudosense similarities, the similarity of the word's pseudosenses was measured using corpus-based distributional similarity and then pseudosenses were paired into a pseudoword based on having similar corpus frequencies and positions along the similarity spectrum. Sense induction models were tested according to their ability to discriminate pseudosenses at different similarity levels. However, the pseudosenses used in their study were not monosemous which limits the ability to replicate their effect in new corpora, which may potentially have different sense distributions of the polysemous pseudosenses. Last, the most similar study is that of Pilehvar and Navigli (2014), which used the same pseudoword dataset as here to analyze supervised and unsupervised WSD. Their findings corroborate those in this study, indicating that the pseudowords of

Pilehvar and Navigli (2013) can be used to design WSD-related evaluations that mirror real-world performance.

7.3 Semi-supervised WSD

Beyond WSID, other approaches have applied semi-supervised learning to WSD. Mihalcea (2004) applies co-training and self-training to the supervised classifier of Lee and Ng (2002), showing that both techniques can reduce the disambiguation error for many words by using high-confidence automatically-labeled examples; however, both techniques required parameter tuning, with no parameter set providing high performance for all words. Pham, Ng, and Lee (2005) investigate four semi-supervised techniques, showing that while spectral graph transduction co-training performed best, performance was not has high as that from purely-supervised WSD methods. Rather than use automatically-labeled data, Yuret (2007) generates new contexts from the existing training data using lexical substitution, which ultimately did not improve in performance when more than a few new substitutes were added. In contrast, other works have seen some improvement over fully-supervised systems using semi-supervised techniques. Niu, Ji, and Tan (2005b) construct a graph of word uses, with edges weighted by the usages' contextual similarity. The annotated instances are labeled in the graph with their senses and then label propagation is run on the graph to infer all remaining instances' labels, with the resulting performance being superior to a SVM-based WSD comparison system. Similarly, Kübler and Zhekova (2009) were able to filter automatically-annotated data based on its expected quality to further supplement the training data. The combination of the manually- and automatically-annotated data provided a slight improvement over the original data, though they note their approach was only able to automatically annotate a small number of contexts per word, illustrating a main challenge for semi-supervised learning of significantly increasing the number of training instances. Martinez, De Lacalle, and Agirre (2008) identify monosemous synonyms of target nouns using WordNet and then query for examples these synonyms on the Web to create a corpus of automatically sense-annotated examples for training. Because WSD performance is closely related to the distribution of word senses, additional heuristics are used to estimate the sense distribution of the testing data (McCarthy et al., 2004) when training a supervised WSD system on the automatically-produced data. The resulting system attained significantly higher performance than unsupervised systems but was still outperformed by some fully-supervised systems.

A common thread of these works is the need for extensive filtering of the unlabeled instances in order to obtain performance improvements; including too many or lower-quality examples typically resulted in performance below supervised technique trained on the same data. In contrast, several of the WSID systems tested here outperformed state of the art without the need for filtering the instances used by the WSI algorithms. The difference in need for filtering suggests that WSI may be more robust to noise in the unlabeled instances –or that WSI could even be a potential preprocessing step for finding the instances most paradigmatic of induced senses for later use as input into semi-supervised techniques.

8. Conclusion

This paper presents a comprehensive analysis of the construction and evaluation of WSID systems. Systems were tested 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 noun senses in WordNet 3.0. In tests on over a million instances, we provide three empirical contributions. First, we demonstrate that the choice of the mapping function used to convert induced senses can significantly affect WSID performance and that a linear kernel SVM significantly improves upon the current state of the art practices (Agirre et al., 2006), with performance increases of 3.8% F1 in some settings. Second, we demonstrate that when using 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 (an 8.5% F1 increase). The benefit of this ensemble setup was further demonstrated in tests on real sense-annotated data using multiple ensemble configurations and different sense inventories, further highlighting its robustness. Third, in a direct comparison with a state of the art supervised WSD system (Zhong & 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. To further this line of research, we have released all implementations of our WSI models and the implementation of our pseudoword testing framework as freely-available open source software at (`https://github.com/osmanbaskaya/mapping-impact`). Furthermore, as a practical result of this effort, we intended to release a large-scale all-words WSID system based on the ensemble model.

The results of this study motivate three interesting avenues for future work that we plan to explore. First, our results indicate that only a few annotated instances are necessary for relatively high WSD performance. Recent work has shown that by controlling for the difficulty that humans have when annotating the contexts (as measured using Passonneau & Carpenter, 2014), the quality of the training data and, subsequently, performance of a WSD system may be improved (Lopez de Lacalle & Agirre, 2015). Together, both findings suggest that high performance WSID systems could be quickly created by appropriately curating the instances to be annotated for training data. In future work, we intend to measure the effect of annotation selection on WSID and examine whether the WSI process itself might also be informative of which instances to select for human annotation.

Second, our experiments were conducted on English-language pseudowords. In future work, we plan to develop analogous pseudoword data for WordNet ontologies in other languages (Bond & Foster, 2013) and replicate these experiments on multilingual data to measure potential language-specific effects when sense-annotated data is sparse. We also plan to investigate using translation and cross-lingual sense mappings to transfer information from English to other languages as a way of gathering the annotations for these WSD systems, analogous to what was done for part of speech tagging (Duong et al., 2014).

Third, the examined WSI models were trained and tested on the multi-domain ukWaC corpus. Typically, WSD has performed much worse when tested on novel domains, which typically contain dissimilar contexts from those in the training data and in which cases, words may have have different dominant senses (Magnini et al., 2002; Preiss & Stevenson, 2013). However, prior works have shown that a small amount of sense-annotation from a novel domain can significantly improve WSD performance in the new domain (Khapra et al., 2010). In future work, we will evaluate whether the WSI system can be used to effectively annotate new instances from the novel domain instead of requiring manual annotation, thus providing an unsupervised method of domain adaptation.

Acknowledgments

We thank Mohammad Taher Pilehvar for many thoughtful discussions and his assistance with the pseudoword dataset. We also thank the reviewers for their comments and suggestions.

References

Agirre, E., de Lacalle, O. L., & Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, *40*(1), 57–84.

Agirre, E., & Martinez, D. (2000). Exploring automatic word sense disambiguation with decision lists and the web. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, pp. 11–19. Association for Computational Linguistics.

Agirre, E., Martínez, D., de Lacalle, O. L., & Soroa, A. (2006). 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*, pp. 89–96. Association for Computational Linguistics.

Agirre, E., & Soroa, A. (2007). Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pp. 7–12. ACL.

Apresjan, J. D. (1974). Regular polysemy. *Linguistics*, *12*(142), 5–32.

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

Baskaya, O., Sert, E., Cirik, V., & Yuret, D. (2013). Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Proceedings of Seventh International Workshop on Semantic Evaluation (SemEval)*, pp. 300–306.

Biemann, C. (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*, pp. 73–80. Association for Computational Linguistics.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Bond, F., & Foster, R. (2013). Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1352–1362.

Brody, S., & Lapata, M. (2009). Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 103–111. Association for Computational Linguistics.

Brody, S., Navigli, R., & Lapata, M. (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 (COLING-ACL)*, pp. 97–104. Association for Computational Linguistics.

Carpuat, M., & Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 61–72. Association for Computational Linguistics.

Chambers, N., & Jurafsky, D. (2010). Improving the Use of Pseudo-Words for Evaluating Selectional Preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Chan, Y., Ng, H., & Chiang, D. (2007). Word sense disambiguation improves statistical machine translation. In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.

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

Dagan, I., Lee, L., & Pereira, F. C. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, *34*(1-3), 43–69.

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

Duan, X., Zhang, M., & Li, H. (2010). Pseudo-word for phrase-based machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 148–156. Association for Computational Linguistics.

Duong, L., Cohn, T., Verspoor, K., Bird, S., & Cook, P. (2014). What can we get from 1000 tokens? a case study of multilingual pos tagging for resource-poor languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 886–897. Association for Computational Linguistics.

Fellbaum, C. (Ed.). (1998). *WordNet: An Electronic Lexical Database*. The MIT Press.

Ferraresi, A., Zanchetta, E., Baroni, M., & Bernardini, S. (2008). Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *In Proceedings of the 4th Web as Corpus Workshop (WAC)*.

Florian, R., & Yarowsky, D. (2002). Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 25–32. Association for Computational Linguistics.

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). Work on statistical methods for word sense disambiguation. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pp. 54–60.

Gaustad, T. (2001). Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Proceedings of the ACL Student Research Workshop*, pp. 61–66. Association for Computational Linguistics.

Graff, D., Kong, J., Chen, K., & Maeda, K. (2003). English Gigaword, LDC2003T05.. Linguistic Data Consortium.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*(suppl 1), 5228–5235.

Hartmann, S., Gurevych, I., & Lap, U. K. P. (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)*. Association for Computational Linguistics.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: the 90% solution. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pp. 57–60. Association for Computational Linguistics.

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, *24*(1), 2–40.

Jurgens, D. (2012). An Evaluation of Graded Sense Disambiguation using Word Sense Induction. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*. Association for Computational Linguistics.

Jurgens, D., & Klapaftis, I. (2013). SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval)*. Association for Computational Linguistics.

Jurgens, D., & Stevens, K. (2011). Measuring the impact of sense similarity on word sense induction. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pp. 113–123. Association for Computational Linguistics.

Khapra, M., Kulkarni, A., Sohoney, S., & Bhattacharyya, P. (2010). All words domain adapted wsd: Finding a middle ground between supervision and unsupervision. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1532–1541. Association for Computational Linguistics.

Kilgarriff, A., & Rosenzweig, J. (2000). Framework and results for english senseval. *Computers and the Humanities*, *34*(1), 15–48.

Kilgarriff, A. (2004). How dominant is the commonest sense of a word?. In *Text, Speech and Dialogue*, pp. 103–111. Springer.

Klapaftis, I. P., & Manandhar, S. (2010). Word sense induction & disambiguation using hierarchical random graphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 745–755. Association for Computational Linguistics.

Kübler, S., & Zhekova, D. (2009). Semi-supervised learning for word sense disambiguation: Quality vs. quantity.. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*.

Lau, J. H., Cook, P., & Baldwin, T. (2013). unimelb: Topic Modelling-based Word Sense Induction. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval)*, pp. 307–311. Association for Computational Linguistics.

Lau, J. H., Cook, P., McCarthy, D., Newman, D., & Baldwin, T. (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 2012)*. Association for Computational Linguistics.

Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 41–48. Association for Computational Linguistics.

Liu, Y., Liu, Z., Chua, T.-S., & Sun, M. (2015). Topical word embeddings. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*.

Lopez de Lacalle, O., & Agirre, E. (2015). Crowdsourced word sense annotations and difficult words and examples. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*.

Lu, Z., Ting, L., & Sheng, L. (2004). Combining neural networks and statistics for chinese word sense disambiguation. In *Proceedings of the Third SIGHAN Workshop on Chinese Language Processing*.

Lu, Z., Wang, H., Yao, J., Liu, T., & Li, S. (2006). An equivalent pseudoword solution to chinese word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (COLING-ACL)*, pp. 457–464. Association for Computational Linguistics.

Magnini, B., Strapparava, C., Pezzulo, G., & Gliozzo, A. (2002). The role of domain information in word sense disambiguation. *Natural Language Engineering*, *8*(4), 359–373.

Manandhar, S., Klapaftis, I. P., Dligach, D., & Pradhan, S. S. (2010). SemEval-2010 Task 14: Word Sense Induction & Disambiguation. In *Proceedings of the Fifth International Workshop on Semantic Evaluation (SemEval)*, pp. 63–68. Association for Computational Linguistics.

Maron, Y., Lamar, M., & Bienenstock, E. (2010). Sphere Embedding: An Application to Part-of-Speech Induction. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., & Culotta, A. (Eds.), *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 1567–1575.

Martinez, D., De Lacalle, O. L., & Agirre, E. (2008). On the use of automatically acquired examples for all-nouns word sense disambiguation.. *Journal of Artificial Intelligence Resesarch (JAIR)*, *33*, 79–107.

Martínez Alonso, H., et al. (2013). *Annotation of regular polysemy: an empirical assessment of the underspecified sense*. Ph.D. thesis, Universitat Pompeu Fabra.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding Predominant Word Senses in Untagged Text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, p. 279, Morristown, NJ, USA. Association for Computational Linguistics.

Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*. Association for Computational Linguistics.

Mihalcea, R., Chklovski, T., & Kilgarriff, A. (2004). The Senseval-3 English Lexical Sample Task. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval)*, pp. 25–28. Association for Computational Linguistics.

Miller, G. A., Leacock, C., Tengi, R., & Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pp. 303–308. Association for Computational Linguistics.

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244.

Nakov, P. I., & Hearst, M. A. (2003). Category-based pseudowords. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pp. 67–69. Association for Computational Linguistics.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, *41*(2), 10.

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

Navigli, R., & Crisafulli, G. (2010). Inducing word senses to improve web search result clustering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 116–126. Association for Computational Linguistics.

Navigli, R., Jurgens, D., & Vannella, D. (2013). Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*.

Navigli, R., Litkowski, K. C., & Hargraves, O. (2007). Semeval-2007 Task 07: Coarse-grained English All-words Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, pp. 30–35. Association for Computational Linguistics.

Navigli, R., & Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 216–225. Association for Computational Linguistics.

Niu, C., Li, W., Srihari, R. K., & Li, H. (2005a). Word independent context pair classification model for word sense disambiguation. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pp. 33–39. Association for Computational Linguistics.

Niu, Z., Ji, D., & Tan, C. (2005b). Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 395–402. Association for Computational Linguistics.

Otrusina, L., & Smrz, P. (2010). A new approach to pseudoword generation.. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*.

Palmer, M., Dang, H. T., & Fellbaum, C. (2007). Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, *13*(02), 137–163.

Passonneau, R. J., & Carpenter, B. (2014). The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, *2*, 311–326.

Passonneau, R., Salleb-Aouissi, A., & Ide, N. (2009). Making sense of word sense variation. In *Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

Pedersen, T. (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 (NAACL)*, pp. 63–69. Association for Computational Linguistics.

Pedersen, T. (2010). Duluth-WSI: SenseClusters Applied to the Sense Induction Task of SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluations*, pp. 363–366.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Petrolito, T., & Bond, F. (2014). A survey of WordNet annotated corpora. In *Proceedings of the Seventh Global Wordnet Conference*, pp. 236–245.

Pham, T. P., Ng, H. T., & Lee, W. S. (2005). Word sense disambiguation with semi-supervised learning. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence (AAAI)*.

Pilehvar, M. T., & Navigli, R. (2013). Paving the way to a large-scale pseudosense-annotated dataset. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1100–1109. Association for Computational Linguistics.

Pilehvar, M. T., & Navigli, R. (2014). A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, *40*(4), 837–881.

Preiss, J., & Stevenson, M. (2013). Unsupervised domain tuning to improve word sense disambiguation.. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 680–684. Association for Computational Linguistics.

Purandare, A., & Pedersen, T. (2004). Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pp. 41–48. Boston.

Rodd, J., Gaskell, G., & Marslen-Wilson, W. (2002). Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*, *46*(2), 245–266.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Schütze, H. (1992). Dimensions of meaning. In *Proceedings of Supercomputing '92*, pp. 787–796.

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

Søgaard, A., & Johannsen, A. (2010). Robust semi-supervised and ensemble-based methods in word sense disambiguation. In *Advances in Natural Language Processing*, pp. 401–405. Springer.

Stevens, K. (2012). Evaluating unsupervised ensembles when applied to word sense induction. In *Proceedings of ACL 2012 Student Research Workshop*, pp. 25–30. Association for Computational Linguistics.

Stokoe, C. (2005). Differentiating homonymy and polysemy in information retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pp. 403–410. Association for Computational Linguistics.

Stolcke, A. (2002). SRILM – An Extensible Language Modeling Toolkit. In *Proceedings International Conference on Spoken Language Processing vol. 2*, pp. 901–904.

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

Tsvetkov, Y., Schneider, N., Hovy, D., Bhatia, A., Faruqui, M., & Dyer, C. (2014). Augmenting english adjective senses with supersenses. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Van de Cruys, T., & Apidianaki, M. (2011). Latent Semantic Word Sense Induction and Disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1476–1485. Association for Computational Linguistics.

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

Wang, J., Bansal, M., Gimpel, K., Ziebart, B. D., & Yu, C. T. (2015). A sense-topic model for word sense induction with unsupervised data enrichment. *Transactions of the Association for Computational Linguistics*, *3*, 59–71.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics (ACL)*, pp. 189–196. Association for Computational Linguistics.

Yuret, D. (2007). Ku: Word sense disambiguation by substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, pp. 207–213. Association for Computational Linguistics.

Yuret, D. (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.

Zhong, Z., & Ng, H. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics.