A Large-Scale Pseudoword-Based Evaluation Framework for State-of-the-Art Word Sense Disambiguation

Mohammad Taher Pilehvar*
Sapienza University of Rome

Roberto Navigli*
Sapienza University of Rome

The evaluation of several tasks in lexical semantics is often limited by the lack of large numbers of manual annotations, not only for training purposes, but also for testing purposes. Word Sense Disambiguation (WSD) is a case in point, as hand-labeled data sets are particularly hard and time-consuming to create. Consequently, evaluations tend to be performed on a small scale, which does not allow for in-depth analysis of the factors that determine a system's performance.

In this article we address this issue by means of a realistic simulation of large-scale evaluation for the WSD task. We do this by providing two main contributions: First, we put forward two novel approaches to the wide-coverage generation of semantically aware pseudowords (i.e., artificial words capable of modeling real polysemous words); second, we leverage the most suitable type of pseudoword to create large pseudosense-annotated corpora, which enable a large-scale experimental framework for the comparison of state-of-the-art supervised and knowledge-based algorithms. Using this framework, we study the impact of supervision and knowledge on the two major disambiguation paradigms and perform an in-depth analysis of the factors which affect their performance.

1. Introduction

Word Sense Disambiguation (WSD) is a core research field in computational linguistics dealing with the automatic assignment of senses to words occurring in a given context (Navigli 2009, 2012). There are two major paradigms in WSD: supervised and knowledge-based. Supervised WSD starts from a training set and learns a computational model of the word of interest, which is later used at test time to classify new instances of the same word. Knowledge-based WSD, instead, performs the disambiguation task by using an existing lexical knowledge base—that is, a semantic network to which graph algorithms, for example, can be applied. However, both disambiguation paradigms have to face the so-called knowledge acquisition bottleneck, namely, the

* Department of Computer Science, Sapienza University of Rome, Viale Regina Elena 295, Roma 00161, Italy. E-mail: {pilehvar,navigli}@di.uniroma1.it.

Submission received: 31 August 2013; revised version received: 31 October 2013; accepted for publication: 6 March 2014.

doi:10.1162/COLI_a_00202

difficulty of capturing knowledge in a computer-usable form (Buchanan and Wilkins 1993).

Unfortunately, providing knowledge on a large scale is a time-consuming process, which has to be carried out separately for each word sense and repeated for each new language of interest. Importantly, the largest manual efforts for providing a wide-coverage semantic network and training corpus for WSD date back to the early 1990s for the WordNet dictionary (Miller et al. 1990; Fellbaum 1998) and to 1993 for the SemCor corpus (Miller et al. 1993). In fact, although cheap and fast annotations could be obtained by means of the Amazon Mechanical Turk (Snow et al. 2008) or voluntary collaborative editing such as in Wikipedia (Mihalcea 2007), producing annotated resources manually is still an arduous and understandably infrequent endeavor. Despite recent efforts in this direction, including OntoNotes (Pradhan et al. 2007b) and MASC (Ide et al. 2010), most work is now aimed either at the automatic acquisition of training data (Zhong and Ng 2009; Moro et al. 2014) and lexical knowledge resources (Navigli 2005; Cuadros and Rigau 2008; Ponzetto and Navigli 2010), or at the large-scale acquisition of annotations via games (Venhuizen et al. 2013) or even video games with a purpose, as recently proposed by Vannella et al. (2014). As a result, state-of-the-art performance can be achieved with both supervised (Zhong and Ng 2010) and knowledge-based (Navigli and Ponzetto 2012b; Moro, Raganato, and Navigli 2014) paradigms in different settings and conditions. Moreover, existing studies hypothesize that this performance can be further improved when larger amounts of manually crafted sense-tagged data or structured knowledge are made available (Martinez 2004; Cuadros and Rigau 2008; Martinez, de Lacalle, and Agirre 2008; Navigli and Lapata 2010). All these results, however, are obtained on small-scale data sets with different characteristics, thus making it difficult to draw conclusions on the factors that impact the system's performance.

In this article we address this issue by providing two main contributions:

- We first focus on novel, flexible techniques for creating new types of artificial words that model real words by preserving their semantics as much as possible. Our semantically aware pseudowords can be used to model any word in the lexicon,[1] therefore aiming for wide coverage. We perform different experiments to show that our semantically aware pseudowords are good at modeling existing ambiguous words in terms of disambiguation difficulty, representativeness, and distinguishability of the artificial senses.
- We leverage our semantically aware pseudowords to create, for the first time, a large-scale evaluation framework for WSD. Using this framework, we are able to perform an experimental comparison of state-of-the-art systems for supervised and knowledge-based WSD on a very large data set made up of millions of sense-tagged sentences. Our large-scale framework enables us to carry out an in-depth analysis of the factors and conditions that determine the systems' performance.

In our recent work (Pilehvar and Navigli 2013), we presented an approach for the generation of semantically aware pseudowords, called similarity-based pseudowords. At the core of this approach was the Personalized PageRank algorithm (Haveliwala

1 Although in our experiments we focus on nouns only, the same approach can potentially be used for any other open-class part of speech.

2002) on the WordNet graph, which was utilized to find the most semantically similar monosemous representative for a given sense of a real ambiguous word. The main strength of the similarity-based approach lies in its flexibility, allowing high minimum frequency constraints to be set on its selection of pseudosenses, while maintaining its overall sense modeling quality.

In this article we extend our previous work as follows: 1) we propose a new approach for generating semantically aware pseudowords which leverages topic signatures; 2) we utilize the best type of pseudoword to create a novel framework for large-scale evaluation and comparison of WSD systems; 3) based on this framework, we carry out a large-scale comparison of state-of-the-art supervised and knowledge-based WSD algorithms; and 4) we study the impact of the amount of supervision and knowledge on the two major disambiguation paradigms and perform an in-depth analysis of the factors and conditions that determine their performance.

The remainder of this article is organized as follows: In Section 2 we survey related work concerning the impact of the knowledge acquisition bottleneck on WSD and provide an explanation of our pseudoword-based approach. In Section 3 we describe pseudowords and overview the existing approaches to their generation. We then present two new approaches that address the issues associated with existing pseudowords, hence enabling the wide-coverage generation of semantically aware pseudowords. In Section 4, we perform various experiments to assess the degree of realism of our proposed pseudowords. We then illustrate how we leverage our pseudowords to generate large sense-tagged data sets in Section 5. The experimental set-up for pseudoword-based WSD is described in Section 6. Experimental results as well as the findings are presented and discussed in Section 7. Finally, we provide concluding remarks in Section 8.

2. Related Work

2.1 Supervised WSD and the Knowledge Acquisition Bottleneck

Over the last few decades, WSD systems have been suffering from disappointingly low performance, especially in an all-words setting in which one has to cover the entire lexicon of the given language (Snyder and Palmer 2004; Pradhan et al. 2007a). In fact, one of the major obstacles to high-performance WSD is the so-called knowledge acquisition bottleneck (Gale, Church, and Yarowsky 1992b): In order to learn accurate word experts, supervised systems need training data for each word of interest, a very demanding task as far as wide coverage is concerned (i.e., one which would require the manual annotation of millions of word instances in context).

In an effort to address this issue, several approaches to the automatic acquisition of sense-tagged corpora have been proposed. Some of these approaches are based on bootstrapping techniques (Yarowsky 1995; Mihalcea 2002; Pham, Ng, and Lee 2005), namely, algorithms which start from a large unlabeled corpus, and a small labeled one, and iteratively populate the latter with an increasing number of sense-annotated sentences from the former data set. Other approaches search the Web or large corpora to retrieve, for each sense, a large number of sentences containing either a set of sense-specific monosemous relatives (Leacock, Chodorow, and Miller 1998; Martinez, de Lacalle, and Agirre 2008) or search phrases (Mihalcea and Moldovan 1999). Collaborative knowledge resources, such as Wikipedia, have also been exploited for generating sense-tagged data (Mihalcea 2007; Shen, Bunescu, and Mihalcea 2013), giving rise to issues, however, such as the encyclopedic nature of the sense inventory and the lack of training of annotators.

An alternative approach to acquiring sense-tagged data is to leverage multilingual resources such as parallel corpora (Chan and Ng 2005a; Wang and Carroll 2005; Chan, Ng, and Zhong 2007). Most of these techniques, however, require human intervention for mapping the translation of a word in the target language to the correct sense of the corresponding word in the source language. Recently, Zhong and Ng (2009) tackled this problem by using a bilingual dictionary. However, the dictionary has to be aligned to the sense inventory of interest (e.g., WordNet) and a large parallel corpus must be available that covers the full range of meanings in a lexicon. The approach, implemented in a system based on Support Vector Machines and called It Makes Sense (Zhong and Ng 2010, IMS), attains state-of-the-art performance on lexical sample and all-words WSD tasks. However, according to our calculation on the available models,[2] this approach can only provide training examples for about one third of ambiguous nouns in WordNet, more than half of which have only one of their senses covered.

Middle-ground approaches have also been proposed that either mix arbitrary sense-tagged corpora with a small amount of tagged data for the domain of interest (Khapra et al. 2010), or estimate the sense distribution of the new domain data set with the help of parallel corpora (Chan and Ng 2005b, 2007), thus relieving the knowledge acquisition bottleneck. However, domain adaptation approaches typically suffer from lower disambiguation performance and still require annotated data for the domain of interest.

2.2 Knowledge-Based WSD and the Knowledge Acquisition Bottleneck

Knowledge-based WSD systems are equally affected by the knowledge acquisition bottleneck, as they exploit the knowledge and structure of lexical knowledge bases in carrying out the disambiguation task. Therefore, in order to obtain high performance, knowledge-based systems are applied to large, wide-coverage lexical knowledge bases. However, the largest hand-crafted resource of this kind (i.e., WordNet) dates back to 1990 with subsequent updates, which attests to the high cost of knowledge engineering on a large scale. Moreover, WordNet mostly provides taxonomic knowledge, while neglecting much syntagmatic relational information between concepts. As a consequence, over the past few years several automatic techniques have been proposed that enrich WordNet with new relation edges, such as those obtained from disambiguated glosses (Mihalcea and Moldovan 2001), collocation dictionaries (Navigli 2005), topic signatures (Agirre et al. 2001; Cuadros and Rigau 2008), and collaborative semi-structured resources (Hovy, Navigli, and Ponzetto 2013).

Enriched knowledge bases have been shown to greatly benefit graph-based approaches such as Personalized PageRank (PPR; Agirre, de Lacalle, and Soroa 2009; Agirre, Lopez de Lacalle, and Soroa 2014), context-based vertex degree (Navigli and Lapata 2010), or, more recently, a densest-subgraph algorithm that jointly performs WSD and Entity Linking (Moro, Raganato, and Navigli 2014). Not only do these methods outperform supervised WSD systems when applied within a domain, but, when the knowledge base is enriched with tens of thousands of semantic relations automatically extracted from Wikipedia, performance comparable to that of state-of-the-art supervised systems can be obtained in a general all-words setting, too (Ponzetto and Navigli 2010; Moro, Raganato, and Navigli 2014).

Recently, a multilingual graph-based WSD approach has been developed that leverages a large multilingual semantic network, called BabelNet (Navigli and Ponzetto

2 http://nlp.comp.nus.edu.sg/sw/models.tar.gz.

2012a), to achieve state-of-the-art results on both general all-words and domain-oriented WSD (Navigli and Ponzetto 2012b). Experimental results show that the joint use of multilingual knowledge enables further improvements over monolingual WSD. However, the power of this disambiguation system lies mainly in its usage of the BabelNet multilingual semantic network. In fact, Agirre, Lopez de Lacalle, and Soroa (2014) showed that under similar conditions (i.e., when the same lexical knowledge base was used), the PPR algorithm can outperform the graph-based WSD algorithms used by Navigli and Ponzetto (2012b).

2.3 The Supervision vs. Knowledge Dilemma

Unfortunately, as of today we do not have unequivocal insights into which disambiguation paradigm is more suitable under which conditions. As a matter of fact, not only does each implemented system come with its own amount and kind of supervision or knowledge, making it hard to determine the contribution of the supervision or knowledge vs. that of the WSD algorithm, but test data sets are small, typically comprising one or two thousand sense-tagged word items, which prevents us from drawing solid conclusions. Even the largest annotation effort ever—namely, the SemCor sense-tagged data set (Miller et al. 1993), comprising around 235,000 semantic annotations—covers only about 15% of word types in WordNet with an average of 10 instances per word, thus precluding large-scale experimental studies.

A possible solution to this current limit in the evaluation of WSD systems is to generate sense-annotated data with the help of artificial ambiguous words, called pseudowords. Pseudowords are created by conflating a set of unambiguous words called pseudosenses. The idea of pseudowords was simultaneously introduced by Gale, Church, and Yarowsky (1992a) and Schütze (1992) as a means of generating large amounts of artificially sense-tagged evaluation data for WSD algorithms. Pseudowords have also been used in other work aimed at studying the effects of data size on machine learning for confusion set disambiguation (Banko and Brill 2001), evaluation of selectional preferences (Erk 2007; Bergsma, Lin, and Goebel 2008; Chambers and Jurafsky 2010), or Word Sense Induction (Di Marco and Navigli 2013; Jurgens and Stevens 2011).

However, constructing a pseudoword by merely combining a random set of unambiguous words picked out to be in the same range of occurrence frequency (Schütze 1992), or leveraging homophones and OCR ambiguities (Yarowsky 1993), does not provide a suitable model of a real polysemous word (Gaustad 2001), since in the real world different senses, unless homonymous, share some semantic or pragmatic relation. For this reason, random pseudowords, when used for WSD evaluation, were found to be easier to disambiguate compared with the human-generated pseudowords (Gaustad 2001), thus leading to an optimistic upper-bound estimate on the performance of WSD classifiers (Nakov and Hearst 2003).

Several researchers addressed the issue of producing pseudowords that can model semantic relationships between senses. To this end Nakov and Hearst (2003) used lexical category membership from a medical term hierarchy (extracted from MeSH[3] [Medical Subject Headings]) to create "more plausible" pseudowords. By considering the distributions from lexical category co-occurrence, they produced a set of pseudowords that were closer to real ambiguous words in terms of disambiguation difficulty than random

3 http://www.nlm.nih.gov/mesh.

pseudowords. However, this approach requires a specific hierarchical lexicon and falls short of creating many pseudowords with high polysemy.

More recent work has focused on the identification of monosemous representatives in the surroundings of a sense, that is, selected among concepts directly related to the given sense. Senses of a real ambiguous word have been modeled by picking out the most similar monosemous morpheme from a Chinese hierarchical lexicon (Lu et al. 2006). Pseudowords are then constructed by conflating these morphemes accordingly. However, this method leverages a specific Chinese hierarchical lexicon, in which different levels of the hierarchy correspond to different levels of sense granularity. A more flexible approach is proposed by Otrusina and Smrz (2010), who model ambiguous words in WordNet. For each particular sense, they search its surroundings in the WordNet graph in order to find an unambiguous representative for that sense.

Unfortunately, as we discuss in detail in the next section, none of these proposals can enable a large-scale evaluation framework for WSD, mainly because they suffer from coverage issues that prevent the creation of wide-coverage sense-annotated data sets. In this article we propose new pseudoword generation techniques that allow for the creation of thousands of artificial words having sufficient occurrence coverage within a large corpus. We then leverage our semantically aware pseudowords to create an evaluation framework which enables a large-scale comparison of state-of-the-art supervised and knowledge-based WSD.

3. Pseudowords

A **pseudoword** is an artificially created ambiguous word created by concatenating two or more distinct words. Formally, $p = w_1{*}w_2{*}\ldots{*}w_n$ is a pseudoword with polysemy degree n where each w_i is called a **pseudosense**. Each pseudosense is usually identified by an unambiguous word drawn from the set of monosemous words in a given lexicon (e.g., WordNet). For instance, *press_release*ship*camel* is a pseudoword with three distinct meanings explicitly identified by its pseudosenses (i.e., *press_release*, *ship*, and *camel*).

Pseudowords are particularly useful for creating artificially annotated data sets. To this end, an untagged corpus C is automatically annotated with a pseudoword $p = w_1{*}w_2{*}\ldots{*}w_n$ by substituting all occurrences of w_i in C with p for each pseudosense $i \in \{1, \ldots, n\}$. As an example, consider the following three sentences:

a1. The goal of a *press release* is to attract favorable media attention.

a2. For a *ship* to float, its weight must be less than that of the water displaced by its hull.

a3. During the winter, the *camel* can go fifty days without being watered.

In order to generate annotated data, it is enough to replace the individual occurrences of *press_release*, *ship* and *camel* with the pseudoword *press_release*ship*camel*, while noting the replaced term as the corresponding sense:

b1. The goal of a *press_release*ship*camel*$_{press\_release}$ is to attract favorable media attention.

b2. For a *press_release*ship*camel*$_{ship}$ to float, its weight must be less than that of the water displaced by the hull.

b3. During the winter, the *press_release*ship*camel*$_{camel}$ can go fifty days without being watered.

where b1, b2, and b3 are three annotated sentences for our pseudoword *press_release*ship*camel* with three different intended senses. This way, pseudowords can be leveraged to automatically annotate an arbitrarily large number of sentences. As mentioned earlier, the first restriction on the choice of pseudosenses is that they need to

be unambiguous, so as to avoid the introduction of uncontrolled ambiguity. Another constraint is that the pseudosense w_i must appear in a sufficient number of sentences in the corpus C. This constraint on the occurrence frequency guarantees that there exist as many sentences in the corpus as the number of annotated sentences that are requested for the task of interest which will exploit the resulting annotated corpus.

The pseudoword in our example was generated by randomly selecting three monosemous words from WordNet. This can be considered as the most immediate approach for generating a pseudoword where constituents are randomly picked from the set of all monosemous words given by a lexicon. This results in a set of pseudowords (hereafter called **random pseudowords**) that are highly likely to have semantically unrelated pseudosenses. However, we know that the different senses of a real word are often in a semantic or etymological relationship. Therefore, random pseudowords can only model homonymous distinctions (such as the *centimeter* vs. *curium* senses of the noun *cm*), and fall short of modeling systematic polysemy (such as the *lack* vs. *insufficiency* senses of the noun *deficiency*).

A pseudoword generation approach ought to be able to address this weakness of random pseudowords. A possible solution is to create pseudowords that model existing ambiguous words by providing, for each pseudoword, a one-to-one correspondence between each pseudosense and a corresponding sense of the modeled word. For instance, *lack*shortfall* is a good pseudoword modeling the real word *deficiency* as its pseudosenses preserve the meanings of their corresponding real word's senses. We call artificial words of this kind **semantically aware pseudowords**, in that they aim at listing senses that are in specific relations to each other, thus mirroring the relations existing between the senses of real words in the lexicon. For example, the lack-insufficiency relation is encoded in the pseudoword for *deficiency*, which would not be possible if we generated a random pseudoword.

Semantically aware pseudowords enable the generation of artificially annotated data sets that have similar properties to their real counterparts and this makes them particularly suitable for the evaluation of WSD and Induction algorithms (Bordag 2006; Jurgens and Stevens 2011; Di Marco and Navigli 2013). In fact, in a real sense-annotated data set different senses of a word appear in distinct contexts. The extent of this distinction, however, depends on the semantic relatedness of the corresponding senses. The intuition behind semantically aware pseudowords is that they model each sense of an ambiguous word through a semantically similar monosemous representative that should appear naturally in contexts that are similar to those of its corresponding real sense. For this reason, these pseudowords should be expected to result in data sets wherein the distinctions between different sense contexts are similar to those in real sense-annotated data sets.

In the next three sections we describe three techniques, two of which are presented in full detail for the first time in this article, for the generation of semantically aware pseudowords that use WordNet as the reference lexicon. In what follows we focus on nominal pseudowords, and leave the extension to other parts of speech to future work.

3.1 Vicinity-Based Pseudowords

As discussed in Section 2, earlier techniques for the generation of semantically aware pseudowords were either inherently restricted to specific hierarchical lexicons utilized in the generation process, or to the number of pseudowords they could generate. An idea put forward by Otrusina and Smrz (2010) was to create pseudowords by combining representatives for each individual sense of a real ambiguous word in WordNet. The

representatives were selected among monosemous relatives (i.e., unambiguous words that are structurally related to a given sense). This method for finding monosemous representatives for senses has been in use since 1998, when it was first proposed for the unsupervised acquisition of sense-tagged corpora (Leacock, Chodorow, and Miller 1998).

Specifically, Otrusina and Smrz (2010) exploit WordNet, whose conceptual units are synonym sets, called synsets, which encode the different meanings of words. In order to find a monosemous representative for a given synset, the approach (hereafter referred to as the **vicinity-based** approach) performs a search on the set of words in the same synset and the surrounding ones (i.e., the synsets connected to that synset by means of WordNet's lexico-semantic relations). These related synsets include siblings and direct hyponyms. In the case where no monosemous candidate could be found among these synsets, the search space is further extended to hypernyms and meronyms.

As an example, consider the ambiguous noun *coke*, which has three senses (i.e., fuel, drink, and drug) in WordNet 3.0. We show in Table 1, for each of the three senses of *coke*, the set of nouns in the corresponding synset as well as in the surrounding synsets. Monosemous words are shown in bold in the table. As can be seen, there exist multiple monosemous candidates for each sense (*coca_cola*, *pepsi*, and *pepsi_cola* for the second sense; *nose_candy*, *cocaine*, and *cocain* for the third sense; and dozens of candidates in the direct siblings' vicinity of the first sense). Among these candidates Otrusina and Smrz (2010) select those whose occurrence frequency ratio in a given text corpus is most similar to that of the senses of the corresponding real word as given by a sense-annotated corpus. However, calculating the occurrence frequency of individual senses of a word requires a large-enough sense-tagged corpus. This dependency on sense-annotated data is a disadvantage of the vicinity-based approach that limits its ability in modeling arbitrary words.

Table 1
Synset neighbors of the three senses of *coke* (We use the sense notation of Navigli [2009] where the i^{th} sense of word w is denoted as w^i. We also highlight monosemous words in bold.)

Synset literals	sense 1	{coke^1}
	sense 2	{**coca_cola**^{1}, coke^2}
	sense 3	{coke^3, blow^6, **nose_candy**^{1}, snow^4, C^{12}}
Direct siblings	sense 1	{biomass^1}, {**butane**^{1}}, {charcoal^1, wood_coal^2}, {**coal_gas**^{1}}, {coke^1}, {**diesel_oil**^{1}}, **diesel_fuel**^{1}}, {fire^7}, {**fossil_fuel**^{1}}, {**fuel_oil**^{1}, **heating_oil**^{1}}, {**gasohol**^{1}}, {**gasoline**^{1}, **gasolene**^{1}, gas^3, **petrol**^{1}}, {**illuminant**^{1}}, {**kerosene**^{1}, **kerosine**^{1}, **lamp_oil**^{1}, **coal_oil**^{1}}, {**methyl_alcohol**^{1}, **wood_alcohol**^{1}, **wood_spirit**^{1}}, {**nuclear_fuel**^{1}}, {**propane**^{1}}, {**red_fire**^{1}}, {**combustible**^{1}, **combustible_material**^{1}}, {**water_gas**^{1}}, {**firewood**^{1}}, {igniter^1, ignitor^1, lighter^1}
	sense 2	{**Pepsi**^{1}, **pepsi_cola**^{1}**}**
	sense 3	{basuco^1}, {crack^8, crack_cocaine^1, tornado^2}
Hypernyms	sense 1	{**fuel**^{1}}
	sense 2	{cola^2, dope^3}
	sense 3	{**cocaine**^{1}, **cocain**^{1}}
Hyponyms	sense 1	-
	sense 2	-
	sense 3	-

Table 2
Noun coverage percentage of vicinity-based pseudowords by degree of polysemy for different values of minimum frequency.

Minimum Frequency	Polysemy 2	3	4	5	6	7	8	9	10	11	12	>12	overall
0	87	82	74	71	67	70	60	64	45	46	44	28	83
50	64	56	47	44	38	41	31	33	17	13	20	10	59
200	52	43	33	27	22	25	19	17	8	10	8	10	46
1,000	31	20	16	7	4	6	4	3	0	0	0	0	25

In addition to this limitation, the vicinity-based approach suffers from lack of flexibility in generating pseudowords that can be leveraged for creating a large-scale pseudosense-tagged corpus, where we need each pseudosense to occur with a relatively large minimum frequency. Due to its small search space, the approach falls short of identifying suitable monosemous representatives for many given senses, which undermines its ability to cover most of the ambiguous nouns in WordNet. We show in Table 2 the percentage of nouns in WordNet that could be modeled using the vicinity-based approach when Gigaword (Graff and Cieri 2003) was used as our corpus. Coverage statistics are presented for four different values of minimum frequency: 0 (no minimum frequency constraint), 50, 200, and 1,000. Besides the overall coverage (rightmost column), in the table we also present the coverage percentage by degree of polysemy. Here, an ambiguous noun in WordNet is considered as covered by its corresponding vicinity-based pseudoword if, for each of its senses, a suitable monosemous candidate can be found in its surrounding that also satisfies the specified minimum frequency in the corpus. As can be seen from the table, the approach can only model about 60% of ambiguous nouns in WordNet 3.0 when a small minimum frequency of 50 sentences in the large Gigaword corpus is assumed. The coverage continues to drop with the increase of minimum frequency up to only 25% of the ambiguous nouns covered when a minimum frequency of 1,000 noun occurrences is required (last row of Table 2), with most of the covered words having low polysemy. This shows that the approach is not flexible enough for generating pseudowords that can be leveraged for creating large, wide-coverage pseudosense-annotated data sets.

In order to address the aforementioned coverage issue of the vicinity-based approach, in the next two sections we propose two new approaches for the generation of semantically aware pseudowords.

3.2 Similarity-Based Pseudowords

We propose a new approach to the generation of pseudowords that enables the creation of semantically aware pseudowords while tackling the coverage and flexibility issues of the vicinity-based approach. In contrast to the vicinity-based method, which takes as its search space the surroundings of a sense, our technique considers the WordNet semantic network in its entirety, hence enabling us to determine a graded degree of similarity between a given sense and all other synsets in WordNet. The **similarity-based** approach identifies, for each sense of a given ambiguous word, the most semantically similar monosemous word satisfying the minimum occurrence frequency constraint. Our method can be considered an extension of the vicinity-based approach as it replaces its pseudosense selection technique with a graph-based similarity measure.

This expands the search space for finding pseudosenses from a small set of surrounding synsets to virtually all synsets in WordNet.

In order to measure semantic similarity we used the PPR (Haveliwala 2002) algorithm, a graph-based technique that has been used previously as a core component for semantic similarity (Hughes and Ramage 2007; Pilehvar, Jurgens, and Navigli 2013) and WSD (Agirre and Soroa 2009; Agirre, Lopez de Lacalle, and Soroa 2014). PPR can be used to estimate a probability distribution denoting the structural importance of all the nodes in a graph for a given target node. When applied on a semantic network, such as the WordNet graph whose nodes are synsets and edges the lexico-semantic relations, the notion of importance can be interpreted as semantic similarity. The reason behind our selecting a graph-based similarity measure was that the alternative context-based methods, such as Lin's (1998) measure, have been shown to require a wide-coverage sense-tagged data set in order to calculate similarities on a sense-by-sense basis for all words in the lexicon (Otrusina and Smrz 2010). Also, among WordNet-based approaches, PPR reports state-of-the-art results on semantic similarity (Agirre et al. 2009) and WSD data sets (Agirre, Lopez de Lacalle, and Soroa 2014), thus representing a suitable graph-based measure for finding the most appropriate pseudosenses.

In Algorithm 1 we present the procedure for the generation of our similarity-based pseudowords. The algorithm takes as input an ambiguous word w, and generates its corresponding similarity-based pseudoword P_w whose i^{th} pseudosense models the i^{th} sense of w. Additionally, the algorithm provides, for each generated pseudoword, a confidence degree denoting the average ranking of the selected pseudosenses.

The algorithm models a given ambiguous word w by iterating over the synsets corresponding to its individual senses (lines 5–18) and identifying the most suitable monosemous representative for each. For each sense of w, we run the PPR algorithm

Algorithm 1: Generate a similarity-based pseudoword

Input: an ambiguous word w in WordNet
Output: a *similarity-based* pseudoword P_w and a confidence score *averageRank*

```
1  begin
2      P_w ← ∅
3      totalRank ← 0
4      i ← 1
5      foreach s ∈ Synsets(w) do
6          similarSynsets ← PersonalizedPageRank(s)
7          sort similarSynsets in descending order;
8          foreach s' ∈ similarSynsets do
9              totalRank ← totalRank + 1
10             foreach w' ∈ SynsetLiterals(s') do
11                 if |Synsets(w')| = 1 and Freq(w') ≥ minFreq and ∄j : (j, w') ∈ P_w then
12                     P_w ← P_w ∪ {(i, w')}
13                     go to line 17
14                 end
15             end
16         end
17         i ← i + 1
18     end
19     averageRank ← totalRank / |Synsets(w)|
20     return (P_w, averageRank)
21 end
```

Table 3
Top five entries of the *similarSynsets* list for different senses of word *coke* (we show both WordNet 3.0 offsets and synsets). The highest-ranking monosemous noun in each list is shown in bold.

Sense no.	Offset	PPR score	Terms in synset (literals)
1	14685768-n	0.225	coke[1]
	14875077-n	0.148	**fuel**[1]
	00498836-v	0.096	coke[4]
	00146138-v	0.038	change_state[1], turn[14]
	15100644-n	0.011	firewood[1]
2	07927931-n	0.237	cola[2], dope[3]
	07928696-n	0.217	**coca_cola**[1], coke[2]
	07927197-n	0.083	soft_drink[1]
	12197601-n	0.045	cola_nut[1], kola_nut[1]
	07928790-n	0.040	pepsi[1], pepsi_cola[1]
3	03060294-n	0.278	**cocaine**[1], **cocain**[1]
	03066743-n	0.205	blow[6], c[12], coke[3], nose_candy[1], snow[4]
	03492717-n	0.046	hard_drug[1]
	00021679-v	0.041	cocainise[1], cocainize[1]
	03060074-n	0.041	coca[3]

by initializing it from the corresponding synset s (line 6). As a result, PPR outputs a probability distribution over all synsets in WordNet denoting the semantic similarity of each synset to s.[4] The synset distribution is then sorted according to its values (line 7). We then go through all its nominal synsets (s') in the search for a suitable monosemous noun (line 11). This search continues until a suitable candidate is found that satisfies the minimum occurrence frequency *minFreq*. Upon finding this candidate, the selected monosemous word w' is added as the corresponding pseudosense for the i^{th} sense of P_w (line 12). These steps are repeated for every sense of w.

The higher the position of a selected pseudosense in the sorted list of *similarSynsets*, the more confidence we have in the preservation of meaning. Therefore, we calculate a confidence score (*averageRank* in the algorithm) as the average of the synset's positions (in the various *similarSynsets* lists) from which the pseudosenses of P_w are picked out (line 19). We will later use this confidence score for evaluating our pseudowords. The algorithm returns as its output, for a given word w, the corresponding pseudoword P_w along with its *averageRank* score (line 20).

Consider the generation process of the similarity-based pseudoword for our word *coke*. Table 3 shows the list of top-five most similar synsets for each of the three senses of this term, as given by the PPR algorithm. Our algorithm selects the highest ranking monosemous candidates that satisfy the minimum frequency (=1,000 in the example) for each sense (shown in bold in the table). Hence, *fuel*coca_cola*cocaine* is returned as the pseudoword corresponding to the word *coke*. Note that the top-ranking synsets are those also found by the vicinity-based approach. However, thanks to PPR working on the entire network, our similarity-based approach can back off to more distant,

4 In the PPR probabililty distribution, the top-ranking synsets contain words that are most likely similar to the target sense, whereas we move to a graded notion of relatedness as far as lower-ranking ones are concerned (Agirre et al. 2009).

Table 4
Sample similarity-based pseudowords generated (with minimum frequency of 1,000 occurrences in Gigaword) for four different nouns in WordNet 3.0. Words shown in bold are those that could not be modeled using the vicinity-based approach for the given minimum frequency. Pseudosenses which are not picked out from the surrounding of the corresponding sense (hence, could not be modeled using the vicinity-based approach) are shown in bold in the second column of the table.

word	similarity-based pseudoword
bernoulli	physicist*mathematician*__astronomer__
coach	football_coach*tutor*passenger_car*clarence*public_transport
green	__greenery__*central_park*labor_leader*green_party*river*golf_course*greens*__max__
sunray	sunbeam*__vine__*sunlight

though similar, synsets. We show in Table 4 some examples of ambiguous words along with their generated similarity-based pseudowords (minimum frequency is again set to 1,000).

Having the large search space of virtually all synsets in WordNet, the similarity-based approach is able to select a monosemous candidate for each sense from a relatively-large ranked list of similar synsets. This solves both the coverage and flexibility issues of the vicinity-based approach for higher values of minimum frequency. However, as mentioned earlier, the higher the position of a selected pseudosense in the sorted list of *similarSynsets*, the more confidence we have in the preservation of meaning. For this reason, we analyzed the *averageRank* values output by Algorithm 1 in order to see how often our algorithm needs to resort to lower-ranking items in the *similarSynsets* list. We present in Table 5, for each polysemy degree and for six different values of *minFreq*, the mean and mode statistics of the *averageRank* scores of all the generated pseudowords for all the nouns (up to polysemy degree 12) in WordNet. As can be seen in the table, the higher the value of *minFreq*, the further the algorithm descends through

Table 5
Statistics of *averageRank* scores of similarity-based pseudowords: we show mean and mode positions for six different values of minimum occurrence frequency (0, 200, 500, 1,000, 2,000, and 5,000) and for each polysemy degree (we show the average value in the case of multiple modes).

minFreq	0		200		500		1,000		2,000		5,000	
poly.	mean	mode	mean	mode	mean	mode	mean	mode	mean	mode	mean	mode
2	2.0	1.0	10.7	2.0	16.9	2.0	28.3	4.0	52.2	3.0	66.8	3.5
3	2.2	2.0	9.7	2.0	15.4	4.7	23.4	4.7	39.6	6.3	51.6	11.7
4	2.3	2.0	8.6	3.0	14.2	7.8	22.5	10.9	33.4	12.3	45.9	18.3
5	2.2	2.0	8.5	5.0	14.6	5.6	21.9	16.0	33.7	14.4	48.3	18.2
6	2.3	2.0	9.0	4.0	15.6	3.8	21.3	12.2	26.5	17.2	41.2	26.0
7	2.2	2.0	8.0	6.0	13.0	8.4	17.8	7.7	26.0	18.9	40.7	27.3
8	2.2	2.0	8.5	4.0	12.7	9.8	19.4	16.1	29.8	28.7	44.1	42.6
9	2.2	2.0	7.5	4.0	12.3	12.4	17.5	17.6	26.4	30.2	40.9	37.9
10	2.2	2.0	7.1	5.0	11.5	7.9	16.2	15.0	25.6	19.2	38.1	38.1
11	2.4	2.0	7.7	8.0	12.0	15.3	16.5	11.5	23.9	7.5	37.9	35.7
12	2.4	2.0	7.7	4.0	12.9	12.9	17.5	23.3	25.7	25.7	44.2	32.5
>12	2.5	1.0	7.2	2.0	10.8	2.0	16.0	4.0	22.4	4.0	39.1	4.0
overall	2.1	1.0	10.1	2.0	16.1	2.0	26.1	4.0	46.3	4.0	60.2	4.0

Table 6
Top five words in the topic signatures for different senses of noun *coke*. The first monosemous noun for each sense is shown in bold.

Sense no.	Sense 1		Sense 2		Sense 3	
	weight	word	weight	word	weight	word
Topic Signatures	0.190	oil	0.477	**pepsi**	0.275	**heroin**
	0.129	gas	0.069	colon	0.151	drug
	0.108	**fuel**	0.031	coca	0.102	hard
	0.092	wood	0.008	coke	0.034	cocaine
	0.079	fire	0.007	star	0.025	user

the list *similarSynsets* to select a pseudosense. However, the mode statistics in the table suggests that even when *minFreq* is set to a large value, most of the pseudosenses are picked out from the highest-ranking positions in the *similarSynsets* list.

3.3 Topic Signature-Based Pseudowords

As an alternative means of finding suitable monosemous representatives for word senses with the PPR algorithm, we propose using automatically generated topic signatures. **Topic signatures** (TS) are weighted topical vectors that are associated with senses or concepts (Lin and Hovy 2000). The dimensions of these vectors are the words in the vocabulary and their weights determine the relatedness of each of these words to the target word sense. These vectors can be obtained automatically from large corpora or the Web with the help of monosemous relatives.

In order to generate a TS-based pseudoword for a word w, we first sort the weighted vectors associated with the senses of w. Then, from each of these vectors, we select the monosemous word with highest relatedness (i.e., largest weight) which satisfies the minimum frequency constraint. The generation process of the TS-based pseudowords is very similar to that of similarity-based pseudowords: Whereas the latter performs a search in the sorted PPR vector of a particular sense to obtain a suitable monosemous representative, the former considers the sorted TS vector as its search space. Also note that the PPR vectors are indexed with synsets, whereas topic signatures have lemmas as their indices.

In our experiments we used the topic signatures provided by Agirre and de Lacalle (2004) for nominal senses of polysemous nouns in WordNet 1.6.[5] The monosemous relatives for each sense were obtained by taking into account WordNet relations such as synonyms, hypernyms, hyponyms, and siblings that were later used to query the Web and create a large corpus. This corpus was then used to build topic signatures.

Table 6 shows the top five words in the topic signatures for different senses of the word *coke*. The first monosemous candidate for each sense is shown in bold (again, the minimum frequency is assumed to be 1,000 here). The corresponding pseudoword generated using this approach is *fuel*pepsi*heroin*.

Even though the TS-based approach shares the monosemous relatives idea with the vicinity-based approach, the additional step of gathering related instances for these representatives guarantees wider coverage. We calculated the coverage of TS-based

5 Available from: `http://ixa.si.ehu.es/Ixa/resources/sensecorpus`.

pseudowords to be 84% (over ambiguous nouns of WordNet 1.6 and with no minimum frequency constraint), which is comparable to that of vicinity-based pseudowords (i.e., 83%, see Table 2). We observed in Table 2 that the coverage of the vicinity-based pseudowords drops rapidly with the increase in minimum frequency such that for a minimum frequency of 1,000 only 25% of the polysemous nouns could be modeled. The TS-based approach, instead, provides a better flexibility for higher values of minimum frequency, hence enabling the generation of large-scale annotated data sets. Thanks to its larger search space, the TS-based approach is able to retain the same 84% coverage for a minimum frequency of 1,000.

Compared with the similarity-based pseudoword generation (described in Section 3.2), this approach provides a different way of overcoming the coverage issue of vicinity-based pseudowords. However, the former guarantees 100% coverage, whereas the latter suffers from the lack of monosemous relatives for a portion of WordNet senses, leading to non-optimal coverage.

4. Pseudoword Evaluation

In Sections 3.2 and 3.3 we presented two techniques for the generation of semantically aware pseudowords that were able to address the coverage and flexibility issues of the vicinity-based approach. In order to verify the ability of these pseudowords to model various properties of real ambiguous words, we performed three separate evaluations so as to assess them from different perspectives:

- Disambiguation difficulty in comparison to real words, where we extrinsically study the impact of the pseudoword quality on the disambiguation performance (Section 4.1).
- Representative power of pseudosenses, where we assess the semantic closeness of pseudosenses to their corresponding real senses (Section 4.2).
- Distinguishability of pseudosenses, where we determine to what extent pseudosenses are specific to a fine-grained real sense rather than covering multiple senses (Section 4.3).

Given that our aim was to leverage these pseudowords for creating large-scale pseudosense-annotated data sets, we performed evaluations on pseudowords generated with *minFreq* per pseudosense set to a high value of 1,000 (i.e., we can generate 1,000 annotated sentences for each pseudosense) using the English Gigaword corpus (Graff and Cieri 2003).

4.1 Disambiguation Difficulty of Pseudowords

Our first experiment is an extrinsic evaluation to assess the correlation between the difficulty of the disambiguation task when using pseudowords and real words. The basic idea behind this experiment is to verify, through a disambiguation task, if the semantic similarity among the senses of an ambiguous word is preserved in its corresponding pseudoword. Semantically similar senses of an ambiguous word will tend to appear in similar contexts, making it relatively difficult to discriminate between them. Conversely, ambiguous words that have semantically distinct senses (e.g., homonyms) will be relatively easier to disambiguate. Given that our pseudowords directly model real ambiguous words, we ideally expect a pseudoword to preserve the same level

of semantic similarity between pseudosenses as that of its corresponding real word, and therefore to exhibit a comparable degree of disambiguation difficulty to that of its corresponding real word.

We performed this evaluation in the style of earlier work (Otrusina and Smrz 2010; Lu et al. 2006). In order to test a pseudoword generation approach using this style, first, all the sense-tagged words in a manually annotated lexical sample data set are modeled using the approach. Next, a corresponding pseudosense-annotated data set is automatically constructed by sampling sentences from a corpus while maintaining the same number of training and test sentences for each word as that of the original manually tagged data set. A correlation analysis is then carried out to compare the disambiguation performance of a supervised WSD system on a given ambiguous word against its corresponding pseudoword. In this experiment we evaluate our similarity-based, TS-based, and, as baseline, random pseudowords. Owing to the fact that for the given minimum frequency of 1,000 we could generate only 5 of the 20 nouns using the vicinity-based approach, we had to exclude the approach from this experiment.

We selected the Senseval-3 English lexical sample data set (Mihalcea, Chklovski, and Kilgarriff 2004) as our manually sense-tagged corpus. The data set provides for 20 nouns of polysemy 3 to 10 an average number of 180 and 90 sense-tagged sentences in its training and test sets, respectively. We generated the similarity-based and TS-based pseudowords corresponding to these 20 nouns, as well as a set of 20 random pseudowords. For each set of these pseudowords we generated corresponding pseudosense-annotated training and test data sets by randomly sampling distinct sentences from the English Gigaword corpus (Graff and Cieri 2003). Therefore, we ended up with four data sets, namely: the Senseval-3 data set of real words, and the three artificially sense-tagged data sets for the similarity-based, TS-based, and random pseudowords. Each of the artificially annotated data sets consisted of training and test portions comprising the same number of instances per sense (i.e., the same sense distribution) as that of the original Senseval-3 training and test data sets. Next, for each of our four data sets, we trained a supervised WSD system on the training set and applied it to the corresponding test set. In order to ensure more reliable results we follow Otrusina and Smrz (2010) and report, for all experiments in this evaluation, the average results for five runs. To this end, we randomly sampled the training and test data sets from the combination of all items while preserving the original proportions. Also, in the random setting, we provide the results averaged on a set of 25 different pseudowords modeling a given ambiguous noun.

As our WSD system for this experiment, we used IMS (Zhong and Ng 2010), a state-of-the-art supervised WSD system that is based on support vector machines (we will describe IMS in more detail in Section 6.4). Note that we measure disambiguation difficulty in terms of the system's recall performance (cf. Section 6.6 for evaluation measures).

We present in Figure 1 the scatterplot of the recall performance (hence the disambiguation difficulty) for real words vs. those for the similarity-based, TS-based, and random pseudowords. For each set of pseudowords, we also show the line fitted to the corresponding set of points by means of linear regression. Ideally, this line should coincide with the dashed diagonal line in the figure, denoting perfect similarity. In the next three subsections we provide an analysis of the scatter plot in Figure 1 and a discussion.

4.1.1 Overall Disambiguation Difficulty. The closer a line of best fit is to the center of the plot, the closer are its corresponding pseudowords to real words in terms of overall

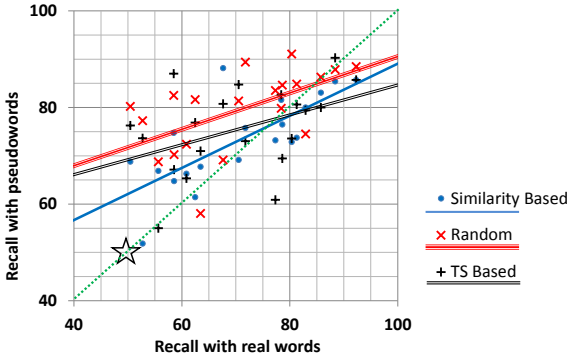

Figure 1
Scatterplot of the recall performance (hence the disambiguation difficulty) of real words versus those for similarity-based, TS-based, and random pseudowords. The star shows the center of the untruncated plot.

disambiguation difficulty (note that the plot's axes are truncated to the range [40,100] and the center point is shown by the star). As can be seen in Figure 1, the line corresponding to our similarity-based pseudowords is the closest to the center, showing that these pseudowords provide a better modeling of real words in terms of disambiguation difficulty. We also show the corresponding values of recall performance in Table 7. We can see from the table that the overall system performance of similarity-based

Table 7
Recall performance of IMS on the 20 nouns of the Senseval-3 lexical-sample test set (Real column) compared with the corresponding similarity-based (SB), TS-based (TS), and random (Rnd) pseudowords. The last three columns show absolute differences between the real setting and the three pseudoword settings.

Word	Real	SB	TS	Rnd	|Real - SB|	|Real - TS|	|Real - Rnd|
argument	50.44	68.79	76.26	77.15	**18.35**	25.82	26.71
arm	92.30	85.69	85.69	88.11	6.61	6.61	**4.19**
atmosphere	70.52	69.15	84.75	80.44	**1.37**	14.23	10.32
audience	81.28	73.74	80.61	83.76	7.54	**0.67**	4.22
bank	85.76	83.07	80.00	82.46	**2.69**	5.76	3.99
degree	78.42	81.58	82.63	80.59	**3.16**	4.21	4.35
difference	62.46	61.43	76.86	75.17	**1.03**	14.40	12.90
difficulty	52.72	51.82	73.64	67.23	**0.90**	20.92	14.97
disc	78.62	76.48	69.45	78.07	**2.14**	9.17	6.18
image	71.78	75.76	73.03	81.50	3.98	**1.25**	10.02
interest	77.34	73.19	60.88	71.70	**4.15**	16.46	6.85
judgment	55.64	66.87	55.00	59.64	11.23	**0.64**	9.01
organization	80.36	72.86	73.57	78.65	7.50	6.79	**3.65**
paper	60.84	66.29	65.33	73.14	5.45	**4.49**	12.59
party	82.94	80.00	79.41	81.04	**2.94**	3.53	3.74
performance	58.56	64.76	67.14	73.86	**6.20**	8.58	15.52
plan	88.42	85.41	90.27	87.39	3.01	**1.85**	3.12
shelter	58.48	74.75	87.00	80.21	**16.27**	28.52	21.73
sort	67.64	88.15	80.74	77.37	20.51	13.10	**9.73**
source	63.46	67.74	70.97	66.26	**4.28**	7.51	7.03
overall	73.26	**75.43**	76.42	78.80	**129.31**	194.51	196.35

pseudowords (75.43) is closest to that of real words (73.26). This value is 76.42 and 78.80 for TS-based and random pseudowords, respectively.

In addition, for 12 of the 20 nouns, the similarity-based approach provides the pseudowords that are closest to real words in terms of WSD recall performance (|Real - SB| column in the table) as shown in bold in the table. This number drops to 5 and 3 for the TS-based and random pseudowords, respectively. Accordingly, the overall sum of the differences (distance) between the recall values is smallest (129.31) for similarity-based pseudowords among the three kinds of pseudoword (194.51 for TS-based pseudowords and an average of 196.35 for random pseudowords, ranging from 158.32 to 262.04).

Even though TS-based pseudowords are only 1% away from similarity-based pseudowords in terms of overall performance, their distance from real words is much higher than that of similarity-based pseudowords (194.51 vs. 129.31). This suggests that the former tend to have a lower correlation with real words than the latter. In the following, we investigate the correlation between the disambiguation difficulties of real words and our three types of pseudowords.

4.1.2 Correlation Between Disambiguation Difficulties. The smaller the angular deviation of the line of best fit for a set of pseudowords is from the diagonal line, the higher is the correlation between the disambiguation difficulties of those pseudowords and real words. As can be seen in the figure, the line corresponding to the similarity-based pseudowords has the smallest deviation from the diagonal line, showing its higher correlation with real words. The Pearson correlation coefficient between the disambiguation difficulties of similarity-based pseudowords and real words is 0.74. This value drops to 0.43 and 0.54 for TS-based and random pseudowords, respectively. Even worse, the value of 0.54 is the average of 25 highly variable correlation values (in the range of [0.18, 0.67]) over our 25 sets of random pseudowords. The reason why TS-based pseudowords show a lower correlation than random pseudowords can be found in the fact that the reported values for the latter are averaged over 25 runs. More precisely, the correlation value of 0.43 of topic signatures has to be compared with the range [0.18, 0.67] of correlations obtained by different sets of random pseudowords.

4.1.3 Discussion. We leveraged PPR and topic signatures as our sense modeling components for the generation of semantically aware pseudowords. Both approaches could solve the low coverage problem, although the results presented in this section suggest that the topic signature-based approach is not good at providing suitable monosemous substitutes for senses of real ambiguous words. A closer look at the similarity-based and TS-based pseudowords generated for some of the nouns in the Senseval-3 data set, shown in Table 8, provides a clear explanation for this shortcoming of topic signature-based pseudowords. In fact, topic signatures are based on co-occurrence information from the Web snippets retrieved for each sense of an ambiguous noun. As a result, many of the top-ranking words in each topic signature are syntagmatically related to the given sense. For example, consider the *paralysis* pseudosense of *arm*, *european* pseudosense of *plan*, and *moral* or *weekly* pseudosenses of *paper*. Despite being semantically related, these pseudosenses cannot be considered as good substitutes for their corresponding senses. Our similarity-based approach, instead, tends to favor paradigmatic (i.e., taxonomic) relations, which is, in fact, the reason behind its better ability at finding suitable substitutes for senses of real words.

Given their significantly lower ability at modeling real words, the TS-based pseudoword generation approach cannot be considered as a candidate for the generation

Table 8
The similarity-based (SB) and TS-based (TS) pseudowords obtained for 6 of the 20 nouns used in our disambiguation experiment.

word	type: equivalent pseudoword
difficulty	SB: workout*deterrent*predicament*complexity TS: get*autism*ski*credibility
arm	SB: forearm*baseball_cap*sword*armchair*executive_branch*garment TS: paralysis*glue*weaponry*wheelchair*pension*clothing
plan	SB: retirement_plan*architect*diagram TS: employee*european*pale
performance	SB: concert*encore*achievement*feat*processing TS: theatrical*musical*ballroom*recruitment*steady
party	SB: political_party*dinner_party*clique*fiesta*someone TS: socialist*prom*transaction*coronation*boomer
paper	SB: piece_of_paper*papers*news_story*telecom*editorial*publishing_house*movie TS: towel*moral*vitamin*brochure*weekly*firm*forecast

of large-scale data sets for our experiments. Hence, we do not consider this type of pseudoword in our further evaluations and focus on similarity-based pseudowords only.

4.2 Representative Power of Pseudosenses

In order to maximize the possibility of preserving the meaning of the original synset, a pseudosense should be selected from the set of words in the same synset, or in the directly related synsets (e.g., hypernym synsets). However, many of the WordNet synsets do not contain monosemous terms and the similarity-based approach often needs to look further into the other indirectly related synsets so as to find a suitable pseudosense. In order to assess how often this happens, we carried out an experiment to get a clear idea of the exact statistics on the distances of the synsets from which pseudosenses are selected from the synsets containing the original senses. To this end, we went through all our similarity-based pseudowords and, for each pseudosense w_i, checked the relationship in WordNet between the synset containing w_i and the corresponding real sense.

We show in Table 9 how the pseudosenses are distributed across different types of WordNet relations, including indirect ones. As can be seen in the table, when *minFreq*

Table 9
Percentage of similarity-based pseudosenses obtained from different types of WordNet relations.

minFreq		0	200	500	1,000
Relation type	Synonyms	24.0	8.3	5.7	4.0
	Hypernyms	30.6	17.9	14.2	11.4
	Hyponyms	9.6	7.2	5.7	4.6
	Meronyms	0.4	0.4	0.3	0.3
	Siblings	9.7	17.9	17.2	16.2
	Other indirect relations	25.7	48.3	56.9	63.5

Table 10
Examples for representativeness scores assigned by the two annotators to pseudosenses of the term *mosaic*.

sense	sense definition (in short) and synset	pseudosense	score (1)	score (2)
1	art consisting of a design made of small pieces {mosaic}	fine_art	3	3
2	viral disease in solanaceous plants {mosaic}	disease	4	3
3	a freeware browser {mosaic}	web_browser	4	4
4	a pattern resembling a mosaic {mosaic}	knowledge	2	1
5	transducer on a television camera tube {mosaic}	electronic_equipment	3	3
6	arrangement of aerial photographs {mosaic, arial_mosaic, photomosaic}	photograph	3	4
	average score		3.17	3.00

is set to 1,000, only about 20% of the pseudosenses are picked out from synonyms or generalization/specialization relations (hypernym and hyponyms). This shows that a considerable portion of our pseudosenses are selected from synsets that are indirectly related to the target synset that is being modeled. These indirectly related synsets can potentially result in pseudosenses that do not have very similar meanings to the original synsets, and hence are not good representatives of them.

Having observed this, we carried out an experiment to evaluate the representative power of similarity-based pseudosenses to assess how well each pseudosense models its corresponding real sense. For this purpose, we randomly sampled 10 pseudowords for each degree of polysemy from 2 to 12 from the entire set of pseudowords[6] generated with minimum frequency of 1,000, totaling 110 pseudowords with 770 pseudosenses. We then asked two annotators, neither of whom was an author of this paper, to judge the representative power of each pseudosense according to the following scale: 1 (completely unrelated), 2 (somewhat related), 3 (good substitute), 4 (perfect substitute). The annotators were provided with the WordNet definitions of the corresponding synsets.

As an example, consider the pseudowords corresponding to the noun *mosaic* shown in Table 10. We present in the table the representativeness scores given by each of our annotators to the individual pseudosenses of this word. The overall representativeness score is calculated as the average of the scores given by the two annotators. In the case of our example, the overall score is 3.085. We also calculated the Spearman correlation between the scores given by the two annotators for all the 770 cases to be 0.66. We show in Table 11 (top) the overall representativeness scores averaged for the full set of 770 pseudosenses, classified by polysemy degree. As can be seen from the table, the overall representative score remains around 3.0 for all polysemy degrees from 2 to 12, with the overall score being 3.1. This shows that even though about 64% of the pseudosenses are picked out from indirect relations (when minimum frequency is 1,000, cf. Table 9), they can still be considered as good representatives for their corresponding real senses. We also present in Table 11 (bottom) the average representativeness scores only for

6 The set contained 15,935 pseudowords corresponding to all the polysemous nouns in WordNet 3.0 (see also Section 6.2).

Table 11
Average representativeness scores for pseudosenses of different polysemy classes (scores range from 1 to 4) and from different WordNet relations. We also show, in the last two rows, the average scores for only those pseudosenses that are picked from synonyms or directly related and sibling synsets.

Polysemy	2	3	4	5	6	7	8	9	10	11	12	overall
Overall score	3.3	3.4	3.1	3.1	2.9	3.1	2.9	2.8	3.3	3.1	3.3	3.1
Direct relations and siblings only	3.4	3.6	3.4	3.3	3.4	3.3	2.8	3.0	3.4	3.2	3.8	3.3
Synonyms only	4.0	–	–	4.0	–	4.0	4.0	4.0	4.0	4.0	4.0	4.0

those pseudosenses that are picked from words in the same synset (synonyms) or in the directly related and sibling synsets. As can be seen, the synonymous pseudosenses are always rated with the highest possible score of 4, whereas those obtained from direct relations maintain a relatively higher score compared with the overall representative score, which includes many pseudosenses picked from indirect relations. In fact, the similarity-based pseudoword generation approach improves the vicinity-based method to full coverage and provides a significantly better level of flexibility for higher values of minimum frequency, while maintaining a good degree of sense modeling ability.

4.3 Distinguishability Between Pseudosenses

A fundamental property of an ambiguous word is that its different senses have distinct meanings. We expect a semantically aware pseudoword to inherit this property of its real counterpart (i.e., to have pseudosenses that are semantically distinguishable from each other while being semantically similar to their corresponding senses). As an example, consider the similarity-based pseudoword *philanthropist*benefactor*[7] corresponding to the noun *donor*.[8] The two pseudosenses of the pseudoword can be considered as good representatives for their corresponding senses. However, the distinguishability of the two real senses is not preserved in the corresponding pseudoword: whereas *philanthropist* only applies to the first sense, *benefactor* can be equally good for both senses of *donor*.

Hence, we performed another evaluation in order to determine the degree of the distinguishability of pseudosenses of our pseudowords. For this evaluation, we used the same set of 110 pseudowords as in the previous experiment (Section 4.2). For each of these pseudowords, we presented its pseudosenses in random order to two annotators. In addition, we provided these annotators with the WordNet definitions of the senses of the corresponding noun and asked them to associate each pseudosense with the most appropriate WordNet sense. The annotators were instructed to leave a pseudosense unmapped if they found it to be equally mappable to multiple senses. We then calculated the distinguishability score for each polysemy degree as the ratio of the number of correct mappings to the total number of senses.

7 From WordNet: "Philanthropist: someone who makes charitable donations intended to increase human well-being"; "Benefactor: a person who helps people or institutions (especially with financial help)."
8 The term *donor* has two senses according to WordNet 3.0: (1) "person who makes a gift of property"; (2) "(medicine) someone who gives blood or tissue or an organ to be used in another person (the host)."

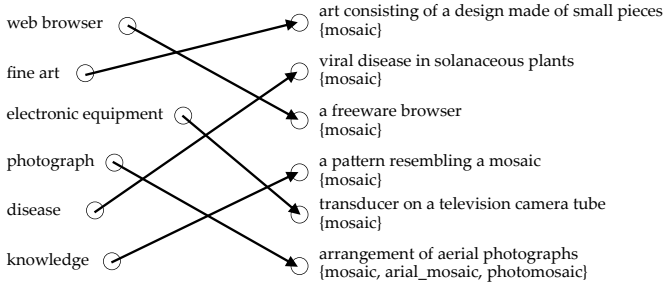

Figure 2
Mappings provided by an annotator from pseudosenses of the similarity-based pseudoword for the noun *mosaic* to its real senses (as defined in WordNet 3.0). In this case all mappings are correct and hence the distinguishability score for this pseudoword will be 6/6=1.

For instance, consider the noun *mosaic* that has six senses in WordNet 3.0. As we also saw in the previous experiment, the corresponding similarity-based pseudoword for this noun is *fine_art*disease*web_browser*knowledge*electronic_equipment*photograph*. As illustrated in Figure 2, we provided the shuffled list of pseudosenses of this pseudoword (left column) and the WordNet definitions of the senses of its corresponding noun, namely, *mosaic* (right column), to each annotator and asked them to map each pseudosense to its most suitable real sense. In this case, both annotators mapped all pseudosenses to their correct senses; hence, the distinguishability score given by each annotator for this pseudoword was 6/6 = 1.

We show in Table 12 the average distinguishability scores for each degree of polysemy 2 to 12 as well as the overall score that is calculated as the average of per-polysemy scores. As can be seen in the table, the distinguishability score is inversely proportional to the polysemy degree (there is a high negative Pearson correlation of 0.9 between the two). However, the score remains above 0.70 even for the pseudowords with higher polysemous degrees. The overall score of 0.79 shows that a large portion of pseudosenses can be associated with their corresponding real senses only. Therefore we can conclude that the similarity-based pseudowords effectively preserve the distinguishability of senses of their real counterparts.

4.4 Discussion

We performed three experiments to evaluate the reliability of our pseudowords. We showed that the similarity-based pseudowords are fairly close to their real counterparts in terms of disambiguation difficulty. Even though our similarity-based pseudowords were slightly easier to disambiguate in comparison to real words, the high correlation observed in the first evaluation (Section 4.1) serves as a guarantee that our pseudowords

Table 12
Average distinguishability scores for pseudosenses of different polysemy classes (scores range from 0 to 1).

Polysemy	2	3	4	5	6	7	8	9	10	11	12	overall
Distinguishability score	0.90	0.83	0.83	0.82	0.81	0.77	0.75	0.73	0.80	0.71	0.70	0.79

can be reliable substitutes for real words in experiments concerning the analysis and comparison of WSD systems.

Our further experiments provided manual evaluations of the representativeness of individual pseudosenses of our similarity-based pseudowords as well as the distinguishability of their pseudosenses from one another. In the representativeness experiment, we assessed, for each individual sense of each pseudoword in our sample set, if the meaning of the corresponding real sense is preserved and if each pseudosense can be considered as a good representative of its corresponding real sense. Finally, in the distinguishability experiment our aim was to investigate the ability of similarity-based pseudowords at preserving the distinguishability among senses of real words. Experimental results proved that the similarity-based approach is able to provide a good modeling of individual senses of real words while preserving the distinguishability of their senses.

5. Sampling Pseudosense-Tagged Corpora

As a result of our evaluations we know that the similarity-based pseudowords are reliable substitutes for real ambiguous words in the disambiguation task. As described in Section 3, a pseudosense-tagged corpus can be generated for each pseudoword $p = w_1 * w_2 * \ldots * w_n$ by substituting individual occurrences of its pseudosenses w_i with the pseudoword p itself, while marking the pseudosense w_i as its annotation. An obvious question that arises here is how to sample and distribute the sentences for a pseudoword across its pseudosenses. In the following two sections we illustrate two corpus sampling strategies used in our experiments.

5.1 Uniform Sense Distribution

A first, simple sampling strategy for pseudosense-tagged corpora is the uniform sense distribution. In this setting, all senses of a pseudoword are assumed to be observed with equal probability in the tagged corpus (i.e., we extract the same number of sentences from the corpus for each pseudosense of a given pseudoword).

5.2 Natural Sense Distribution

Although the uniform distribution can be useful in specific applications such as dictionary disambiguation (Litkowski 2004; Flati and Navigli 2012), or knowledge resource mapping (Navigli and Ponzetto 2012a; Matuschek and Gurevych 2013), in natural text we know that most of the occurrences of an ambiguous word correspond to a usually small subset of predominant senses of that word (Zipf 1949; Sanderson and Van Rijsbergen 1999). In other words, occurrences of an ambiguous word in a real text are usually distributed across its senses according to a highly skewed distribution. In order to model this natural distribution, we adopt a distribution sampling strategy. To this end we estimate sense distributions from SemCor (Miller et al. 1993), the largest sense-tagged corpus of English. However, as we show in Table 13, SemCor provides reliable distribution estimates for only some hundred words. The table shows for each degree of polysemy the number of distinct nouns in SemCor that are sense-annotated at least once, 10 times, or 20 times, compared with the corresponding total number of ambiguous nouns in WordNet (last column in the table). Because we could obtain from SemCor the sense distribution of only some hundred

Table 13
Number of distinct nouns annotated in SemCor at least 1, 10, or 20 times. We also show the total number of WordNet ambiguous nouns (last row) for different polysemy degrees.

Polysemy	2	3	4	5	6	7	8	9	10	11	12	>12	total
Frequency ≥ 1	2,349	1,315	703	453	241	183	81	91	57	45	23	49	5,590
Frequency ≥ 10	301	242	196	180	122	97	61	60	43	35	20	43	1,400
Frequency ≥ 20	122	97	112	111	70	63	47	42	29	24	15	35	767
WN amb. nouns	10,257	2,989	1,178	620	306	212	96	94	60	48	25	50	15,935

Table 14
Average sense distribution for nouns in SemCor. We select only those nouns for which there exist at least 10 sense-tagged occurrences in SemCor.

Poly.	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
2	87.8	12.2										
3	78.1	17.9	4.0									
4	72.9	19.6	6.4	1.1								
5	71.1	18.8	7.4	2.3	0.4							
6	65.4	21.5	8.5	3.1	1.2	0.3						
7	61.1	23.0	9.5	4.1	1.8	0.4	0.1					
8	62.7	20.9	9.3	4.4	1.8	0.6	0.2	0.1				
9	56.0	22.4	10.4	5.6	3.4	1.6	0.6	0.0	0.0			
10	51.1	24.4	11.7	7.0	3.6	1.4	0.6	0.2	0.0	0.0		
11	50.7	23.5	12.0	6.8	3.6	1.7	1.0	0.4	0.2	0.1	0.0	
12	54.3	18.2	9.1	7.2	4.4	2.7	2.1	1.4	0.5	0.1	0.0	0.0

low-polysemy nouns and a few dozen high-polysemy nouns, we decided to drop the requirement of estimating sense distributions directly (i.e., to model the semantically aware pseudoword p_w on the sense distribution of w). Instead, we first collected all the sense distributions of nouns with at least 10 occurrences in SemCor. Our choice of 10 as the minimum occurrence frequency was to guarantee some hundreds of distributions for lower polysemy degrees and dozens for the higher ones (see Table 13). In addition, given the highly skewed nature of sense distributions in SemCor, 10 samples should usually be enough for a reliable estimation of the corresponding sense distributions to be made, even for higher polysemy degrees. Having at hand a large set of distributions estimated for each polysemy degree, every time we needed a new pseudoword with m senses, we randomly picked out a sense distribution of size m from our collection.

We show the macro-averaged[9] sense distribution for each polysemy degree from 2 to 12 in Table 14. As can be seen from the table, all average distributions, especially those of low polysemy nouns, are skewed towards predominant senses.

9 There are some outliers in SemCor that would have negatively affected the average distributions if micro-averaging was used. For instance, the three-sense word *person* has over 6,000 instances, all of which are tagged with the first sense. This would bias the micro-averaged sense distribution given that there exist approximately an overall 17,000 instances for three-sense nouns. For our sampling strategy, though, we do not need distribution averaging.

6. Experimental Set-up

In this article up to this point we have provided the basis for creating large-scale pseudosense-annotated data sets by proposing a flexible approach for generating semantically aware pseudowords that model arbitrary real words. We have also explained different sampling strategies for distributing pseudosense-annotated sentences according to two different distributions. We are now ready to set up our experimental framework for large-scale WSD.

We first describe the text corpus used in our experiments (Section 6.1), then explain how we selected a reliable subset of pseudowords for the experiments (Section 6.2); this is followed by a description of the process of generating training and test data sets (Section 6.3). In Section 6.4 we introduce the two WSD systems used as representatives of the two main WSD paradigms (i.e., supervised and knowledge-based) in our experiments. We then provide, in Section 6.5, the details of the method through application of which our knowledge-based system is able to benefit from the training data. Finally, in Section 6.6 we describe the evaluation measures used in our experiments.

6.1 Corpus

We sampled all the sentences for pseudosense tagging from the English Gigaword corpus (Graff and Cieri 2003), a comprehensive corpus of English newswire text. The corpus comprises about 4.1 million documents, each containing an average of 430 words, totaling approximately 1.76 billion words. In a preprocessing phase, we removed sentences whose length was either longer than 50 words or shorter than 10 words. The corpus was then annotated with part-of-speech tags using the C&C tagger (Curran and Clark 2003) trained on the Penn Treebank (Marcus et al. 1994). The resulting corpus contained around 50 million sentences.

6.2 Pseudoword Selection

As a result of our similarity-based approach, we could generate as many pseudowords as polysemous nouns in WordNet 3.0 (i.e., 15,935 pseudowords). However, for two reasons that will be explained shortly, we only considered a reliable subset of these pseudowords for generating the data sets for our experiments.

Firstly, we did not consider nouns with polysemy degree higher than 12 in our experiments, as it is not possible to perform a reliable analysis on such degrees given that very few pseudowords can be generated for them (about 0.3% of ambiguous nouns in WordNet have polysemy degree 13 or higher). Secondly, we observed that in practice a large enough portion of pseudowords for each polysemy degree can provide a reliable performance estimation on that polysemy degree. Therefore, we selected, for each polysemy degree, the top 300 pseudowords according to the calculated *averageRank* score (cf. Section 3.2). Given that the score denoted our confidence in the preservation of meaning while modeling pseudosenses, this top-ranking subset of pseudowords is the most reliable one. Table 15 (first row) shows the distribution of this subset of pseudowords across different degrees of polysemy. Note that for polysemy degrees 6 to 12, where there exist less than 300 nouns in WordNet, we consider all the corresponding pseudowords. In Appendix A, we show that this subset is large enough for an accurate estimation of the performance of a WSD system. We also sampled a separate set of 199 pseudowords for tuning purposes (cf. Section 6.5.1 for tuning). Table 15 (second

Table 15
Number of pseudowords per degree of polysemy (2 to 12) in our test and tuning sets.

Polysemy	2	3	4	5	6	7	8	9	10	11	12	Total
Test set	300	300	300	300	278	192	84	87	54	43	22	1,960
Tuning set	30	30	30	30	28	19	9	9	6	5	3	199

row) shows the distribution of this tuning set of pseudowords across different polysemy degrees.

6.3 Generating Data Sets

Data set size. The first question that comes to mind before generating data sets is that of the number of sentences to be pseudosense-tagged for each pseudoword. As we showed in Section 3.2 (Table 5), the minimum occurrence frequency (*minFreq*, which corresponds to the number of sentences to be tagged with a particular pseudosense) directly affects the *averageRank* score, a measure that we interpreted as our confidence in the preservation of meaning of a real sense through its corresponding pseudosense. This suggests a trade-off between the scale of our experiments and their overall accuracy. We show in Table 16 the statistics of the *averageRank* score for the subset of pseudowords selected for our experiment when six different values of *minFreq* were assumed while generating pseudowords. Currently, the MASC corpus (Ide et al. 2010), even though covering a small set of 20 nouns, provides the highest number of manually annotated instances per word, namely, 1,000 sentences. In our experiments we followed MASC and generated 1,000 annotated instances for each of our 1,960 pseudowords. As can be seen from Table 16, when *minFreq* = 1,000, a pseudosense is on average selected from the 8.4^{th} position in the *similarSynsets* list (given by mean), with most of them being picked out from the first position (given by mode).

Table 16
Statistics of the *averageRank* score of the subset of pseudowords selected for our experiments: we show mean and mode statistics for six different values of minimum occurrence frequency (*minFreq*) and for each polysemy degree (average value is presented in the case of multiple modes).

minFreq	0		200		500		1,000		2,000		5,000	
poly.	mean	mode	mean	mode	mean	mode	mean	mode	mean	mode	mean	mode
2	1.0	1.0	1.0	1.0	1.1	1.0	1.2	1.0	1.5	1.0	1.9	2.0
3	1.0	1.0	1.6	2.0	2.3	2.7	2.8	3.2	3.5	4.0	5.4	7.7
4	1.0	1.0	2.4	3.0	3.8	5.3	5.2	7.1	6.8	7.4	11.6	14.8
5	1.4	1.0	3.8	5.0	6.2	5.6	8.8	13.6	12.5	14.4	20.8	18.2
6	2.1	2.0	6.4	4.0	10.2	3.8	14.8	12.2	20.8	17.2	35.8	26.0
7	2.0	2.0	6.4	6.0	10.1	8.4	14.0	7.7	20.3	18.9	33.7	27.3
8	1.9	2.0	6.5	4.0	10.1	9.8	15.9	13.5	23.9	21.9	37.9	42.6
9	2.0	2.0	6.3	4.0	10.7	12.4	15.4	17.6	22.6	30.2	36.0	37.9
10	2.0	2.0	5.7	5.0	9.2	7.9	13.3	15.0	20.9	19.2	31.9	31.9
11	2.1	2.0	6.8	8.0	10.8	13.6	13.9	11.5	20.0	7.5	32.0	35.7
12	2.2	2.0	5.6	4.0	10.5	10.5	14.3	23.3	21.7	21.7	37.0	32.5
overall	1.5	1.0	3.8	1.0	6.0	1.0	8.4	1.0	11.9	2.0	19.7	2.0

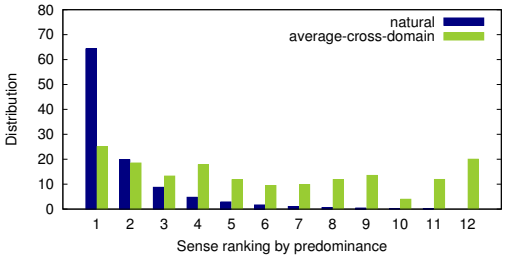

Figure 3
Sense distribution in a naturally distributed text as well as the average of sense distributions across different domains.

Data set configurations. In addition to being large-scale and accurate, we also wanted our experiments to cover a wide range of possible real-world scenarios. In Section 5, we identified two different sense distributions according to which we could produce pseudosense-tagged corpora, namely, the uniform distribution and the natural one. In our experiments, we considered all four possible ways of combining the sense distributions of training–test data—that is, *Natural-Natural* (Nat-Nat), *Uniform-Uniform* (Uni-Uni), *Uniform-Natural* (Uni-Nat), and *Natural-Uniform* (Nat-Uni). We provide the following rationale for each of them:

Nat-Nat. This is the traditional open-text WSD scenario (Kilgarriff and Rosenzweig 2000; McCarthy et al. 2004), in which senses are naturally distributed according to the same distribution both in the training and the test data sets.

Nat-Uni. This configuration trains a WSD system with a natural distribution but applies it to texts for which the distribution is unknown (e.g., in different domains). Given that any choice of a different sense distribution for the test data set would have been arbitrary, we selected the uniform one as the approximate average of sense distributions across different domains. In other words, our assumption was that the uniform distribution could be thought of as the fairest different distribution. To verify this, we studied the variability of sense distribution across texts belonging to different domains. We started from a data set of sense-annotated documents from 30 different domains provided by Faralli and Navigli (2012). We then estimated the average sense distribution of all nouns across documents, shown in Figure 3 (light columns) for polysemy degrees 2 to 12 as sorted according to WordNet sense order. As can be seen, the average sense distribution across domains is not skewed, in contrast to the natural sense distribution (dark columns in the figure). In addition, this configuration models a setting in which the system is not effectively provided with knowledge of all senses in the test set.

Uni-Uni. This configuration assumes a system with the same amount of knowledge for all senses, tested on a task in which all senses are equally important. As is also the case for the Nat-Uni configuration, the uniformly distributed test set in Uni-Uni also models tasks such as dictionary disambiguation, in which sense-wise precision matters (Flati and Navigli 2012).

Uni-Nat. Similarly to Uni-Uni, this configuration takes no stand on the training sense distribution, but tests it on naturally distributed data.

Data set split. We created our training and test sets by sampling 1,000 sentences per pseudoword from the Gigaword corpus for each of the two sense distributions (i.e., natural and uniform). Out of these sentences, we kept 200 (i.e., 20%) as a test set and used the remaining 800 (i.e., 80%) for training. In order to be able to analyze the impact of the amount of knowledge on the disambiguation performance, the 800 sentences in the training data set were split into 10 different subsets of varying size (from 80 sentences [i.e., 10% of training instances] to the full set of 800 sentences in 10 steps) while at the same time preserving the original sense distribution for all these sets. Overall, the data sets comprised about 2 million[10] pseudosense-tagged sentences for each of the four configurations.

6.4 Systems

We chose state-of-the-art off-the-shelf representatives for the two mainstream WSD paradigms, that is, supervised and knowledge-based WSD.

6.4.1 Supervised: It Makes Sense (IMS). In our experiments we used IMS (Zhong and Ng 2010) as the representative supervised WSD system. IMS is a publicly available English all-words WSD system achieving state-of-the-art results on several Senseval and SemEval tasks.[11] The system classifies words in context using linear support vector machines. The context (a sentence in our case) is represented as a standard vector of features including parts of speech, surrounding words, and local collocations (Lee and Ng 2002).

For each of the four configurations (see Section 6.3) and for each pseudoword, IMS was trained with the corresponding training set and the learned word expert model was then applied to the test set. In our experiments, we used the default configuration of IMS where the system adopts a linear SVM classifier with L2-loss function.

6.4.2 Knowledge-Based: UKB. As the state-of-the-art knowledge-based WSD system, we used UKB.[12] UKB is a publicly available graph-based WSD system that exploits a pre-existing lexical knowledge base (Agirre, Lopez de Lacalle, and Soroa 2014). UKB provides an implementation of the PPR algorithm (Haveliwala 2002), adapted to the task of WSD, as proposed by Agirre and Soroa (2009). PPR is applied to a graph representation of a Lexical Knowledge Base (LKB), which is typically WordNet or an extension of it with additional semantic edges. We used the *w2w* variant, which has been shown to perform best (Agirre and Soroa 2009), where PPR is initialized by concentrating the probability mass on the context words other than the target word to be disambiguated. The most suitable sense of the latter is then chosen by selecting the highest-ranking vertex (i.e., sense) of the word.

Similarly to IMS, we used for UKB the corresponding training set in each training–test configuration. However, a typical knowledge-based WSD system (such as UKB) cannot directly learn from the training data (which, instead, is naturally suited to supervised WSD systems). In the following section we describe the method used in our experiments to transfer these data into readily available knowledge for UKB.

10 1,000 sentences × (1,960+199) pseudowords.
11 http://nlp.comp.nus.edu.sg/software/ims/.
12 http://ixa2.si.ehu.es/ukb/.

Hereafter, we will use IMS and UKB to mean supervised and knowledge-based systems, respectively, since we consider these two systems as state-of-the-art representatives of their corresponding paradigms.

6.5 Enriching the LKB Using Training Data

Whereas supervised WSD exploits a training set to perform sense classification, knowledge-based approaches use lexical knowledge bases instead. Therefore, a similar operation to that of providing an increasingly large training set is to enrich basic knowledge bases such as WordNet with additional semantic edges, as has previously been done, among others, by Navigli (2005), Cuadros and Rigau (2008), and Navigli and Lapata (2010). The automatic knowledge injection step, however, is less immediate and natural than the supervised one. In fact, pseudowords cannot be directly used to obtain a ready-to-use set of relation edges. To cope with this issue, in each configuration we used the corresponding training set (on which IMS was trained) to extract knowledge that could be used to enrich the WordNet LKB.[13] To this end, given a pseudoword p and for each pseudosense $w_i \in p$, we identified the most semantically related words w' to w_i using the Dice coefficient:

$$\frac{2c(w_i, w')}{c(w_i) + c(w')} \tag{1}$$

where $c(w_i, w')$ is the number of sentences in which w_i and w' co-occur, and $c(w_i)$ and $c(w')$ are the total number of sentences containing individual occurrences of w_i and w', respectively. We then connect, in the WordNet graph, w_i to all the senses of each of the top-K related words. Ideally, the corpus used for calculating these statistics should be fully sense-tagged, namely, each usage of an ambiguous co-occurring word tagged with the intended sense. However, because our training data (as is customary for WSD lexical sample data sets) do not provide sense annotations for context words, these edges are semi-noisy in that we connect an unambiguous endpoint w_i to all senses of w'.

As an example, consider the pseudosense $w_i = airplane$, which is directly linked to 40 other nodes (synsets) in WordNet: 15 hyponyms, 10 meronyms, 14 domain-related synsets, and a hypernym. We show 10 of these connections in Figure 4 (dashed lines). By exploiting the sentences tagged with pseudosense *airplane* in the training set, we obtain the list of K top-ranking semantically related words using the above-mentioned procedure. We then connect in the WordNet graph $airplane_n^1$ to every sense of these words (we show 10 such new linkings in the figure). The highlighted nodes in the figure are the new direct neighbors of $airplane_n^1$ in the enriched graph. As can be seen in the example, our enrichment approach provides many additional syntagmatic relations to the initial mostly paradigmatic relations in WordNet. As a result of this enrichment procedure, we are able to generate a LKB consisting of WordNet plus semi-noisy semantic edges obtained from the co-occurrence statistics of each pseudosense of our pseudowords.

6.5.1 Tuning. Although we described the method used in our experiments to obtain new semantic edges with the help of co-occurrence statistics, we did not show how we set the value of K—that is, the number of top-ranking related words we obtain from a given set of n pseudosense-tagged sentences to be used for LKB enrichment.

13 We used the WordNet 3.0 LKB provided in UKB.

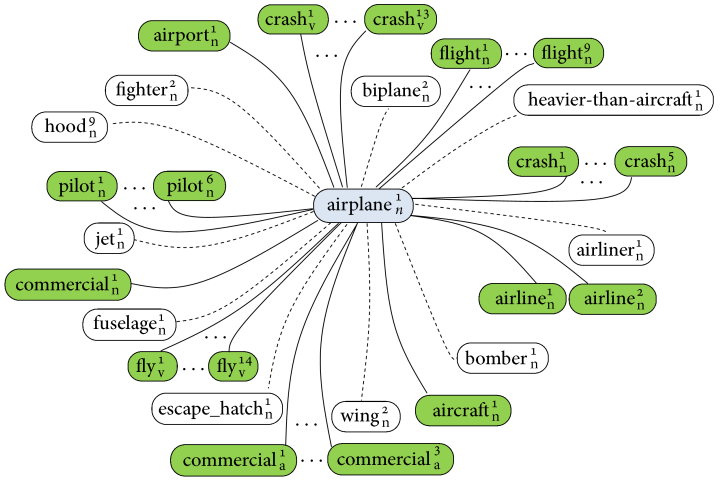

Figure 4
Enriching the WordNet LKB by adding edges between *airplane* and its top-K most similar words (as obtained from the corresponding training set). For brevity, we show only a small part of nodes here (*airplane* is connected to 40 synsets in WordNet). The dashed lines correspond to existing edges in WordNet and solid lines represent the new additional edges. The highlighted nodes are the new direct neighbors of *airplane* in the enriched LKB graph.

To calculate the optimal value of K, we carried out a tuning experiment on a data set built for our subset of 199 pseudowords dedicated for tuning (cf. Section 6.2). In order to consider the data set size factor in our tuning, we experimented on three different sizes of training data: 80, 400, and 800 sentences (first, middle, and last size steps). For each of these training data set sizes, we generated LKBs for different values of K and carried out disambiguation on the tuning test set.

Figure 5 shows how the UKB recall performance (for more details on our evaluations measure see Section 6.6) varies when the value of K is varied from 25 to 800 (in increasing steps of 25). We show in the figure the average performance value for the three training sizes (i.e., 80, 400, and 800). As can be observed from the figure, recall

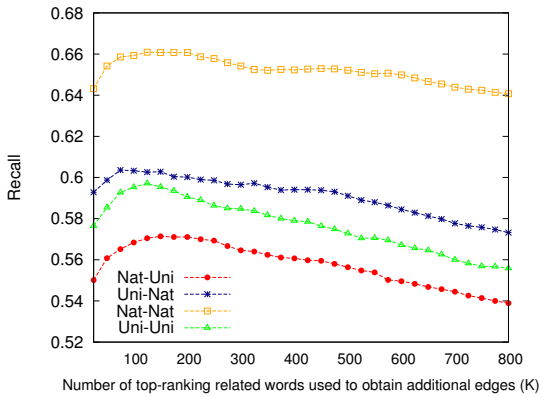

Figure 5
UKB recall performance when varying the number of top-ranking related words used for the LKB enrichment (K) from 25 to 800 (in 32 steps). We show the average performance over three sizes of training data: 80, 400, and 800 sentences.

is not always directly proportional to the number of additional edges. In fact, after a certain point recall starts to decay as the number of additional edges increases. We present in Table 17 the corresponding values for a part of Figure 5 (i.e., K in the range [25, 300]), where the optimal recall value seems to occur for all four data set configurations. As can be seen in the table, the best performance occurs at $K = 125$ for Uni-Uni and Nat-Nat configurations. However, the best performance is seen at $K = 75$ and $K = 150$ for Uni-Nat and Nat-Uni configurations, respectively.

Given that for the supervised system we did not perform any tuning based on the sense distribution of the test data set, in order to enable a fair comparison we chose the same optimal K value irrespective of the test data. The last two rows in the table show the average performance for the two distributions of training data (for instance, "Uni-*" stands for the average performance over Uni-Uni and Uni-Nat configurations). It can be seen that the maximum average performance occurs at $K = 125$ for the Uni training data and at $K = 150$ for the Nat training data. Therefore, depending on the sense distribution of training data, we used two different cutting thresholds (K) on the number of related words considered for enriching the corresponding LKB.

6.6 Evaluation Measures

It is customary in the WSD literature to evaluate the performance of a disambiguation system based on precision, recall, and F1 measure (Navigli 2009). Precision calculates the portion of items that are correctly disambiguated from among the total output by the system, and recall measures the portion of the total items in the data set that are correctly disambiguated by the system. F1 is the harmonic mean of precision and recall. Because in our setting all the pseudowords to be disambiguated in the test set are covered in the training data and also included as a node in the LKB, IMS and UKB always provide an answer for each item in the test set. For such a full-coverage case, the values of precision, recall, and F1 will be equal. Hence, in our experiments, we report the recall performance of the systems only. In addition, throughout this article, we present the results in terms of recall percentage (i.e., the value of recall multiplied by 100).

7. Experiments and Results

As discussed in the experimental set-up, WSD experiments were carried out with IMS and UKB while injecting an increasingly higher amount of supervision and knowledge, respectively, that is, from 0 to 800 training sentences (cf. Section 6.3). We show the overall

Table 17
Recall performance of UKB when varying K for all the four data set configurations. The last two rows show the average performance for each training data distribution. The maximum values for each configuration are shown in bold.

edges (K)	25	50	75	100	125	150	175	200	225	250	275	300
Uni-Uni	57.65	58.54	59.27	59.53	**59.71**	59.54	59.35	59.06	58.90	58.63	58.50	58.48
Nat-Nat	64.32	65.42	65.86	65.93	**66.09**	66.07	66.07	66.07	65.87	65.78	65.58	65.42
Uni-Nat	59.28	59.87	**60.36**	60.33	60.26	60.28	60.04	60.02	59.90	59.86	59.68	59.65
Nat-Uni	55.01	56.08	56.52	56.84	57.05	**57.14**	57.11	57.11	57.00	56.93	56.67	56.46
Uni-*	58.47	59.20	59.82	59.93	**59.99**	59.91	59.69	59.54	59.40	59.24	59.09	59.06
Nat-*	59.66	60.75	61.19	61.38	61.57	**61.61**	61.59	61.59	61.43	61.35	61.13	60.94

Table 18
Performance of IMS and UKB on the naturally distributed test set when varying the size of the training set per pseudoword (MFS for Nat-Nat = 70.5%). Note that we do not report an MFS baseline for Uni-Nat configuration as there is no most frequent sense in the training data.

Config.	Train	System	Size of training data										
			0	80	160	240	320	400	480	560	640	720	800
Nat-Nat	Nat	IMS	–	**81.9***	**84.4**	**86.4**	**87.4**	**88.2**	**88.7**	**89.3**	**89.7**	**90.0**	**90.3***
		UKB	38.8	56.5$^{\diamond}$	59.2	61.1	62.1	62.8	63.4	63.8	64.2	64.4	64.6$^{\diamond}$
Uni-Nat	Uni	IMS	–	**59.8**	**66.3**	**69.8**	**72.2**	**74.0**	**75.2**	**76.3**	**77.2**	**77.9**	**78.6**
		UKB	38.8	53.6	55.2	56.4	57.5	58.4	59.1	59.7	60.1	60.5	60.8

Table 19
Performance of IMS and UKB on the uniformly distributed test set when varying the size of the training set per pseudoword (MFS = 25.0%). Note that for Uni-Uni the MFS baseline is not affected by the choice of the most frequent sense in the training data.

Config.	Train	System	Size of training data										
			0	80	160	240	320	400	480	560	640	720	800
Nat-Uni	Nat	IMS	–	35.7*	39.0	41.1	42.5	43.8	44.6	45.4	46.0	46.6	47.1*
		UKB	38.8	**52.3**$^{\diamond}$	**54.4**	**55.7**	**56.5**	**57.1**	**57.4**	**57.8**	**58.1**	**58.4**	**58.6**$^{\diamond}$
Uni-Uni	Uni	IMS	–	**59.8**	**66.4**	**70.0**	**72.5**	**74.3**	**75.5**	**76.6**	**77.5**	**78.2**	**78.9**
		UKB	38.8	53.9	55.4	56.5	57.6	58.6	59.3	59.9	60.3	60.7	61.0

recall performance of both systems on natural and uniform test sets in Tables 18 and 19, respectively.[14] Note that in each table the training set can also be either uniformly or naturally distributed, resulting in an overall four training–test configurations for each system in the two tables. For each configuration, we show the recall performance values as we vary the size of the corresponding training set from 0 to 800 sentences per pseudoword (whereas the size of the test set, which comprises 200 sentences per pseudoword, is the same across different training sizes, cf. Section 6.3). For 0 training size, we only show the results of UKB, which is merely based on the vanilla WordNet LKB.

The Most Frequent Sense (MFS) baseline values for the naturally and uniformly distributed test sets are 70.5% and 25.0%, respectively. The best recall performance (among the two systems) in each training–test configuration and for each size of the training data set is shown in bold.

7.1 General Overview of the Results

We observe that IMS has a considerably larger performance variation across different configurations (ranging from 35.7%* to 81.9%* with 80 training sentences, and from 47.1%* to 90.3%* with 800), whereas UKB is less sensitive to training and test distributions (52.3%$^{\diamond}$–56.5%$^{\diamond}$ with 80 training sentences, and 58.6%$^{\diamond}$–64.6%$^{\diamond}$ with 800). The

14 Symbols in the tables are for easier referencing.

performance of IMS (Nat-Nat) with 160 training sentences is in line with competitive results on the Senseval-3 lexical sample data set (Mihalcea, Chklovski, and Kilgarriff 2004) in which there exist around 180 training sentences for each noun on average. In fact the latter are in the 73% ballpark against an MFS recall of 55.2% (Zhong and Ng 2010), whereas IMS obtains 84.4% against 70.5% MFS in our setting. The 15% shift in MFS is due to the sense distribution of our Nat data set, which is more skewed towards frequent senses compared to that of the Senseval-3 lexical sample data set. In addition, the average polysemy degree of our pseudowords is slightly lower than that of the Senseval-3 nouns (average polysemy 5.1 of pseudowords vs. 5.8 of Senseval-3 nouns), which also contributes to higher recall in our experiments.

7.2 Corroboration of Previous Findings on a Large Scale

Before moving to a detailed analysis and discussion of our results, we briefly report here on the results of the experiments that were conducted in order to confirm some of the previous findings in the literature on a large scale. We provide the details of these experiments in Appendix B. In summary, we were interested in verifying:

- The relation between system's performance and polysemy degree: our results confirm previous findings by Palmer, Dang, and Fellbaum (2007) that the two are inversely proportional. We also found IMS to be particularly robust on highly polysemous words in the Nat-Nat configuration. The inverse proportionality was approximately logarithmic in all system configurations except for IMS in the Nat-Nat configuration in which the proportionality was linear (Appendix B.1).
- The relation between connectivity of a node in the LKB and disambiguation accuracy: we found that the higher the connectivity of a node, the more accurate will be its disambiguation. This corroborates the preliminary findings of Navigli and Lapata (2010) on a much larger scale (Appendix B.2).

Previous work has made claims only on significantly smaller amounts of annotated data (e.g., in Navigli and Lapata 2010), whereas we show for the first time that these hold in large-scale experiments with several orders of magnitudes more annotated data.

7.3 UKB Largely Benefits from Semi-Noisy Edges

Thanks to our framework, we can go into considerably greater detail on the second point that we verified in Section 7.2. As can be seen in Tables 18 and 19, the enrichment of the WordNet LKB proves to be highly beneficial. The performance of UKB increases significantly, even when the edges are harvested from a low number of training sentences, which is particularly impressive because the added edges are semi-noisy. In fact, recall grows, with Nat test, from 38.8% to 56.5% (Nat-Nat) and 53.6% (Uni-Nat), and, with Uni test, from 38.8% to 52.3% (Nat-Uni) and 53.9% (Uni-Uni), when using just 80 training sentences per pseudoword. The impact of semi-noisy edges is even higher with more training data, ranging between +19.8% and +25.8% improvement when the full set of 800 training sentences is used.

7.4 Performance of the Systems in Different Configurations: IMS Leads

Except in Nat-Uni, IMS outperforms UKB in all other configurations. The gap is particularly evident in the Nat-Nat configuration, which is the typical setting for lexical sample WSD tasks. The performance of UKB is closer to that of IMS for smaller sizes of training data irrespective of the configuration. The gap, however, expands with the growth in the number of sentences in the training set. This shows that the learning rate of IMS is faster than that for UKB.

Nat-Uni is the only configuration in which UKB surpasses IMS. The low performance of IMS shows that providing enough training instances for all senses is always beneficial for IMS, and this happens in the Nat-Nat, Uni-Uni, and Uni-Nat configurations.

IMS benefits from two advantages in the Nat-Nat configuration: (1) It is aware of the sense distribution in the test set and (2) Due to the skewed sense distribution in this configuration, often some of the senses of a word are not covered in the training data (and in the test set as well). This reduces the number of classes in the classification task, making it a potentially easier task to carry out. We will talk more about the first point in Section 7.6. The second point, namely, the partial coverage of senses, even though making the disambiguation an easier task in the Nat-Nat configuration, is responsible for the very low performance of IMS on the uniformly distributed test set (i.e., Nat-Uni configuration). UKB, on the other hand, is not as sensitive as its supervised counterpart to the sense distribution, making it robust across different configurations—with generally lower performance, however.

7.5 UKB is More Robust with Respect to Sense Skewness

To investigate if our WSD systems are biased in favor of more frequent senses, we calculated the recall performance by sense predominance in the Nat test setting (in the Uni test setting we assume no sense predominance). In other words, we separately calculated the recall performance of IMS and UKB on items tagged in the test set with the first (i.e., most frequent) sense of each pseudoword, the second (i.e., second most frequent) sense, and so on. We show in Figure 6 the overall recall by sense predominance, averaged over the 10 training sizes for each of the two possible configurations with the Nat test set. As can be seen UKB tends to be more robust

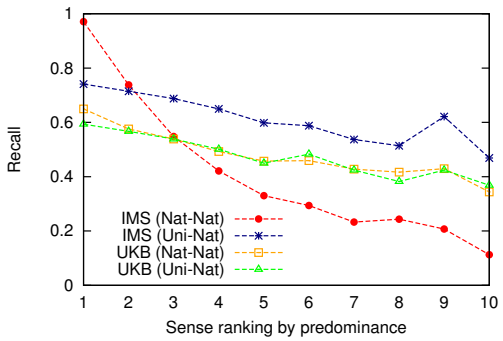

Figure 6
Recall by sense predominance in the naturally distributed test set (we show up to 10th order, since higher orders do not have many instances to provide reliable results).

across sense ranking, irrespective of the distribution of the training data. In contrast, IMS is not equally robust across configurations: Although its recall is relatively stable in the Uni-Nat configuration, it is not when the training set is naturally distributed (Nat-Nat). In fact, this shows that IMS, when trained on naturally distributed data, is biased towards classifying most of the instances as more frequent senses. This lack of robustness is shown in the figure from the rapid performance drop of IMS in the Nat-Nat configuration (from 97.1% recall on the most predominant sense to about 11.3% on the tenth pseudosense of a pseudoword), indicating that the system tends to perform considerably better on more frequent senses when a natural distribution is assumed.

7.6 Impact of the Knowledge of Sense Distribution

Previous work (Escudero, Màrquez, and Rigau 2000; Agirre and Martínez 2004; Chan and Ng 2005b) has highlighted the impact of the underlying sense distribution for a supervised disambiguation task. However, we do not know much, especially on a large scale, about the effect of integrating sense distribution information into knowledge-based systems. In order to gain more insight into this, we carried out a pilot study to see how much improvement UKB can gain if explicitly provided with domain knowledge in the form of sense distribution, so as to give the same advantage to both knowledge-based and supervised systems. UKB provides, for each disambiguation instance, a probability distribution over the senses of the target word where each probability value can be regarded as the chance of the corresponding sense to be selected in the given context. A possible way to inject the sense distribution knowledge into UKB is to scale the scores assigned to each sense by the corresponding probability values in the sense distribution. According to this procedure, the scaled score P_i for the i^{th} sense of a target word w (with $|S|$ senses) is obtained by:

$$P_i = \frac{p_i d_i}{\sum_{s \in S} p_s d_s} \tag{2}$$

where d_i is the probability of the i^{th} sense according to the corresponding sense distribution and p_i is the probability score assigned to the i^{th} sense of w by UKB in the given context. In this way we provide UKB with additional information that can increase the selection chance of more frequent senses in situations wherein the system is not confident in its choice of the winning sense (i.e., instances where UKB considers comparable chances for multiple senses to be selected). We used two different ways of calculating sense probability values d_i:

- Pseudoword-specific sense distribution, where the probability values are directly calculated from the corresponding pseudoword that is being disambiguated.
- Average distribution, where values of d_i are the average per polysemy probabilities estimated using SemCor (cf. Table 14).

We show in Table 20 the overall UKB performance in the Nat-Nat configuration when injected into the pseudoword-specific and average sense probabilities. We can see from the table that, when using only the WordNet LKB (training size = 0), the provided pseudoword-specific information can boost the performance of UKB by over 26 percentage points (from 38.9 percentage points to 65.6 percentage points). For other

Table 20
Performance of UKB on Nat-Nat configuration when injected with the pseudoword-specific (specific) and average sense distribution (average) information as well as the original performance values for UKB and IMS (as reported earlier in Table 18). All the improvements of the UKB (specific) and UKB (average) systems over the UKB (original) system are statistically significant at the $p < 0.001$ level.

Size of training data	0	80	160	240	320	400	480	560	640	720	800
IMS	-	81.9*	84.4	86.4	87.4	88.2	88.7	89.3	89.7	90.0	90.3
UKB (original)	38.8	56.5	59.2	61.1	62.1	62.8	63.4	63.8	64.2	64.4	64.6
UKB (specific)	65.6	78.0*	78.8	79.2	79.5	79.7	79.9	80.1	80.3	80.4	80.4
UKB (average)	58.7	75.2	75.8	76.3	76.7	76.8	76.9	77.0	77.1	77.2	77.2

sizes of the training data an improvement of 15 percentage points to 21 percentage points is achieved. When provided with this additional sense distribution information, UKB yields performance comparable to that of IMS (especially for smaller sizes of the training data, e.g., 78.0* of UKB vs. 81.9* of IMS for training size 80). Note that, being a supervised system, IMS already benefits from this pseudoword-specific sense distribution information. Similarly, when provided with the average sense distribution information, UKB exhibits a considerable improvement ranging from 20 percentage points for WordNet LKB only (training size = 0) to 12 percentage points for the training size of 800 sentences.

7.7 Performance Upperbound of the Systems

A possible way to examine how well a system fits the training data is to carry out training and test on the same data set. This is precisely the setting we explore in this experiment. Our aim was to have an estimate of the performance upperbound of each of our two WSD systems. We observed that IMS attains an optimal recall value of 100.0 for all data set sizes and for both sense distributions (i.e., uniform and natural distributions), showing that its models perfectly fit the training data. However, as mentioned earlier in Section 6.5, in our setting the automatic enrichment of the LKB is less immediate and natural than the training of the supervised system. In fact, the annotated data cannot be directly utilized by UKB. Instead, co-occurrence statistics obtained from this data were used to enrich the LKB of UKB. We estimated the performance upperbound of UKB (and therefore the capability of our LKB enrichment approach) by performing an experiment where additional edges were obtained by exploiting the sentences in the test data set. The enrichment procedure was, however, the same as the one used in our main experiments (see Section 6.5). The experiment was carried out on both our test data sets, namely, naturally and uniformly distributed. In addition, we did not use the values of K (i.e., maximum number of related words per pseudosense used for enriching the LKB) tuned for our main experiments (cf. Section 6.5.1) as we expected the test sentences to be able to provide more beneficial additional edges; instead we used five different values of K, from 200 to 1,000.

Table 21 shows the UKB performance when additional edges are obtained from test data sets. We present the overall as well as polysemy-specific performance values for five different values of K. As expected, a higher performance was shown by UKB in this setting in comparison with the normal setting where sentences in the training data were exploited for enriching the LKB. An interesting finding here is that even when the test

Table 21
Performance of UKB when additional edges are obtained by exploiting the sentences in the test data set. We show results for both uniformly and naturally distributed test data and for different values of *K* (maximum number of related words per pseudosense used for enriching LKB).

		Uniformly distributed data set					Naturally distributed data set				
	K	200	400	600	800	1,000	200	400	600	800	1,000
	overall	78.6	83.8	86.4	87.9	88.4	78.0	82.4	85.5	86.6	88.4
By polysemy	2	87.9	91.7	93.2	94.2	94.9	89.4	92.1	93.9	93.7	95.1
	3	83.6	87.5	89.6	91.3	92.0	83.0	87.1	89.5	90.4	92.2
	4	79.8	84.5	87.3	88.9	89.5	79.1	83.6	86.8	87.7	89.5
	5	78.1	82.9	85.7	87.3	87.7	76.9	81.8	84.9	85.6	87.9
	6	75.1	81.1	84.2	85.8	86.9	74.3	79.3	83.0	84.7	86.4
	7	73.4	80.3	83.4	84.7	85.0	72.2	77.1	80.8	81.6	84.2
	8	71.0	78.1	80.5	81.8	82.1	69.6	74.2	78.0	82.4	82.5
	9	70.5	77.3	80.8	82.6	82.5	68.0	73.0	77.3	78.5	80.9
	10	68.0	75.1	77.8	79.2	79.7	67.0	72.5	76.8	78.2	80.2
	11	70.3	77.8	80.5	81.4	81.5	69.2	75.3	79.7	83.1	82.4
	12	64.6	73.3	77.0	78.8	79.4	68.2	73.5	77.5	80.3	81.9

data set is used for obtaining the additional edges, UKB can hardly cross into 90%. In other words, there is a gap of about 10% resulting from the semi-noisy enrichment of UKB. This shows that our LKB enrichment approach is not optimal. We defer the task of improving the current enrichment technique to future work. In fact, our framework enables other knowledge enrichment approaches to be effectively tested and compared.

7.8 Summary of the Results

Here, we summarize our experiments aimed at analyzing the behavior of state-of-the-art supervised and knowledge-based systems in different settings, and also to verify their dependence on various factors:

- IMS has a considerably larger performance variation across different data set configurations whereas UKB is less sensitive to the underlying sense distribution (Section 7.1).
- The semi-noisy enrichment of LKB results in a huge improvement in the performance of UKB even when the edges are harvested from a low number of training sentences (Section 7.3).
- IMS outperforms UKB in all configurations but Nat-Uni, which models a WSD setting across different domains where some senses in the test set might not be covered in the training data set. The gap between IMS and UKB is especially noticeable in the Nat-Nat configuration (Section 7.4).
- UKB is more robust with respect to sense skewness whereas IMS is highly biased towards classifying most of the instances as more frequent senses (Section 7.5).
- Injecting sense distribution information to UKB highly boosts its performance, providing an interesting mixture of knowledge and supervision (Section 7.6).

- The upperbound performance of IMS when trained on the test data set is 1.0 whereas that of UKB lags 0.1 behind (Section 7.7).

In addition, we performed a set of experiments in order to verify some existing findings in the literature at a large scale. We briefly reported these results in Section 7.2. See Appendix B for the details.

8. Conclusion

In this article we proposed a novel framework for the experimental comparison of state-of-the-art supervised and knowledge-based WSD systems on a large scale. At the core of our approach lies the usage of a new type of realistic pseudowords, which makes it possible to model virtually all ambiguous nouns in a lexicon. As a result, we could generate pseudowords modeling each polysemous noun in WordNet, whose high quality we assessed from different perspectives.

We selected a reliable subset of pseudowords for each of which we sampled 1,000 tagged instances from a large corpus, resulting in a 2 million pseudosense-tagged corpus. This corpus was then used for training a state-of-the-art supervised WSD system (i.e., IMS) as well as for automatically injecting large quantities of (semi-noisy) semantic relations into the WordNet graph for use by an off-the-shelf knowledge-based WSD system (i.e., UKB). Our pseudoword-based framework enabled the analysis of the conditions and factors that impact the performance of these state-of-the-art WSD systems on a large scale, a study which has never heretofore been possible.

We hope our work will pave the way for new research on the generation and exploitation of large-scale sense-annotated corpora. Furthermore, our new type of pseudoword might also be used for a realistic, wide-coverage evaluation of other difficult tasks such as Word Sense Induction (Bordag 2006; Di Marco and Navigli 2013; Navigli and Vannella 2013), Entity Linking (Moro, Raganato, and Navigli 2014) and selectional preference acquisition (Chambers and Jurafsky 2010; Erk, Padó, and Padó 2010), among others.

We are releasing to the research community the entire set of 15,935 pseudowords of WordNet 3.0 polysemous nouns, including those selected for our WSD experiments (`http://lcl.uniroma1.it/pseudowords/`). Together with the pseudosense-annotated corpus, this will allow for future experimental comparisons and studies with other WSD systems, also in other languages. In fact, our pseudowords and our WSD framework are not language-dependent and can readily be applied to other languages with the help of multilingual semantic networks such as BabelNet (Navigli and Ponzetto 2012a) and the use of multilingual WSD algorithms (Moro, Raganato, and Navigli 2014). Finally, along the lines of Cuadros and Rigau (2007), our framework could be used in the future to test and compare various LKB enrichment techniques.

Appendix A: Reliability of Our Findings

In order to verify if the selected subset of pseudowords (cf. Section 6.3) was large enough to provide a reliable estimation of per-polysemy performance, we calculated, for both our systems, confidence intervals of performance values. Table A.1 shows, for the training set consisting of 400 sentences, the 95% confidence interval values for the obtained per-polysemy performance (Table B.1) as well as for the overall performance (Tables 18 and 19). As could have been expected, the confidence interval is smaller for lower polysemy degrees where there exist more pseudowords. We can see from

Table A.1
Two-sided 95% confidence interval for the performance values when the training set contains 400 sentences for both the systems and for all our four configurations.

Polysemy		2	3	4	5	6	7	8	9	10	11	12	overall
# of pseudowords		300	300	300	300	278	192	84	87	54	43	22	1,960
Nat-Nat	IMS	0.82	0.89	0.94	0.94	0.97	1.21	1.87	1.92	2.45	2.20	5.48	0.41
	UKB	2.00	2.17	2.21	2.18	2.25	2.60	3.74	3.74	4.87	4.11	7.67	0.96
Uni-Uni	IMS	0.74	0.92	0.85	0.85	0.91	1.06	1.42	1.42	1.80	2.06	2.81	0.51
	UKB	1.30	1.29	1.40	1.35	1.37	1.77	2.55	2.65	2.86	3.04	4.68	0.69
Uni-Nat	IMS	0.80	1.15	1.08	1.21	1.19	1.51	2.21	2.51	2.73	2.90	6.20	0.59
	UKB	2.26	2.47	2.41	2.33	2.34	2.76	4.01	4.00	4.76	4.13	7.95	1.00
Nat-Uni	IMS	1.87	1.86	1.62	1.53	1.38	1.58	2.28	2.06	2.44	2.34	3.97	0.79
	UKB	1.47	1.41	1.32	1.31	1.35	1.73	2.30	2.49	2.60	2.81	4.42	0.68

the table that the confidence interval always remains below 2.0 and 3.0, respectively, for IMS and UKB for polysemy degrees up to five for which we set an upperbound of 300 pseudowords. This shows that the subset of 300 pseudowords we picked for those polysemy degrees is large enough to provide an accurate polysemy-specific performance.

In addition, the overall confidence interval is always $\leq$ 1.0 in all system configurations. This shows that the overall recall performance values we reported in Tables 18 and 19 are quite accurate. These results also hold for other sizes of the training data.

Appendix B: Corroboration of Previous Findings

In the next two sections, we provide the details of the experiments carried out in order to verify some existing findings in the literature on a large scale.

Table B.1
Average per polysemy performance of IMS and UKB in all the four configurations (averaged over all the 10 size steps which is also very close to the results with 400 sentences).

	Nat-Nat		Uni-Uni		Uni-Nat		Nat-Uni		MFS baseline	
polysemy	IMS	UKB	IMS	UKB	IMS	UKB	IMS	UKB	Nat-Nat	* – Uni
2	94.7	81.7	87.5	76.5	87.5	78.8	64.4	74.0	87.7	50.0
3	91.0	70.8	80.0	66.6	79.5	66.1	51.8	64.5	78.2	33.3
4	88.9	63.5	75.6	59.5	75.9	59.0	44.7	57.6	72.0	25.0
5	87.6	60.7	71.8	55.4	71.7	56.1	38.5	53.6	71.3	20.0
6	85.9	55.3	68.1	52.6	68.7	51.3	35.6	51.7	64.2	16.7
7	83.4	52.5	65.0	49.1	65.2	48.0	33.9	47.7	58.8	14.3
8	82.2	49.3	62.2	46.2	60.7	46.0	30.6	45.2	59.0	12.5
9	81.2	47.1	60.3	44.9	58.9	42.9	28.8	44.1	58.7	11.1
10	78.5	42.6	58.1	42.0	55.1	39.2	27.9	41.4	52.4	10.0
11	76.1	46.3	53.5	43.8	53.7	43.2	27.0	42.2	49.6	9.1
12	75.2	44.6	52.5	39.7	51.9	42.4	26.2	38.8	54.9	8.3

B.1 Performance by Polysemy

Previous work (Palmer, Dang, and Fellbaum 2007) has shown that both manual and automatic disambiguation can be affected by polysemy. In this section, we verify with our large-scale framework the relation between disambiguation performance and polysemy degree. Table B.1 presents the performance values (averaged over all 10 sizes of training data) as classified by polysemy degree for each system and for all the four configurations. As a general trend, irrespective of the configuration and system, the performance is inversely proportional to polysemy degree. The type of inverse proportionality is approximately logarithmic in all system configurations except for IMS in the Nat-Nat configuration, where it is approximately linear.

Another interesting observation is in the variation of the polysemy-wise performance difference between the two systems across different configurations. On average, the absolute polysemy-wise difference between the two systems is 14.1 in the Uni-Uni, Uni-Nat, and Nat-Uni configurations with the minimum difference being 8.7 (Uni-Nat, polysemy 2) and the maximum being 17.4 (Uni-Nat again, polysemy 6). However, in the Nat-Nat configuration the difference between the two systems increases rapidly with polysemy. Starting with a value of 13 at polysemy 2, the difference value rapidly increases with polysemy to a maximum of 35.9 at polysemy 10 (the absolute difference is on average 28.2 in this configuration). This divergence in the polysemy-wise performance of our two systems in the Nat-Nat configuration shows that IMS, in addition to being particularly good at this configuration, is able to further extend its lead over UKB at higher polysemy degrees.

B.2 Performance by Pseudosense Node Degree

As discussed in Section 6.4, UKB adopts the PPR algorithm, a variant of eigenvector centrality, whose behavior highly depends on the structure of the graph it is applied to. Previous research (Cuadros and Rigau 2006; Navigli 2008; Navigli and Lapata 2010) has shown that a denser graph with a large number of semantic relations benefits the eigenvector centrality-based approaches, enabling them to provide more accurate disambiguation judgments. These evaluations, however, were carried out on the WordNet graph leveraged for the disambiguation of instances from the SemCor data set. In this section, we perform a similar analysis but on a much larger scale, that is, in a setting with hundreds of thousands of disambiguation instances and using a much denser graph. Essentially, the graphs used in our experiments consist of the same nodes (i.e., synsets) as the WordNet graph but enriched with thousands of additional semantic edges obtained from co-occurrence statistics (cf. Section 6.5).

We follow Navigli and Lapata (2010) and take as our measure of graph connectivity the degree centrality which is calculated based on the number of edges incident to a particular node in a graph. We show in Figure B.1 how the nodes are distributed in the graph according to their degree. We present the distributions for the two LKBs enriched with full naturally and uniformly distributed training data (i.e., 800 sentences) as well as for the original WordNet graph. The slightly higher degree of the nodes in the LKB enriched using the naturally distributed data set is due to the availability of a higher number of additional edges per pseudosense in this setting (obtained from 150 related words per pseudosense for the naturally distributed data set vs. 125 for the uniformly distributed data set, cf. Section 6.5.1).

In Figure B.2, we show the average node degree for different ranges of UKB recall performance (20 intervals from 0.0 to 1.0). Each point in the graph shows the average

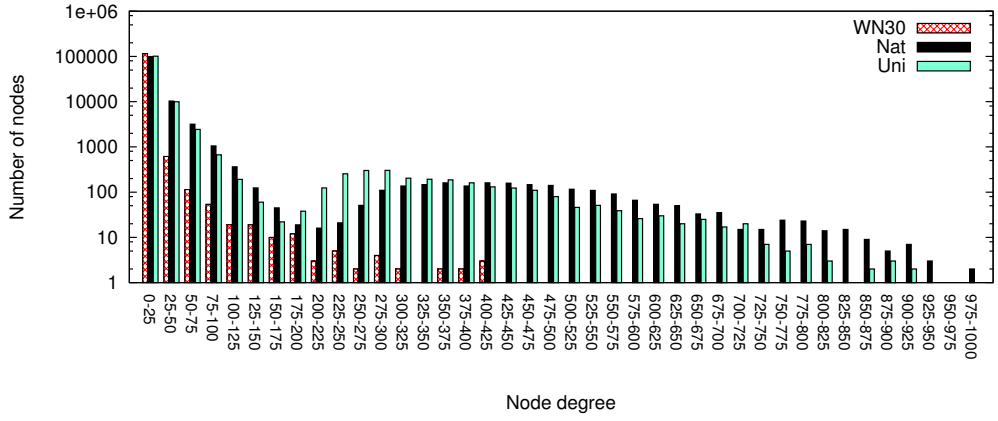

Figure B.1
Distribution of nodes by incident edges in three different LKBs: WordNet 3.0 (WN30), the enriched LKB with 800 sentences of uniformly- (Uni) and naturally- (Nat) distributed data sets.

degree of the set of nodes (i.e., pseudosenses) on which UKB obtains a recall that falls within the corresponding range. As can be seen in the figure, irrespective of the configuration and training size, the higher the connectivity of a node, the more accurate will be its disambiguation. This is in line with earlier research (Navigli and Lapata 2010), in which it is hypothesized that WSD performance increases when the target sense in the graph tends to have a higher number of incident edges. On the other hand, this

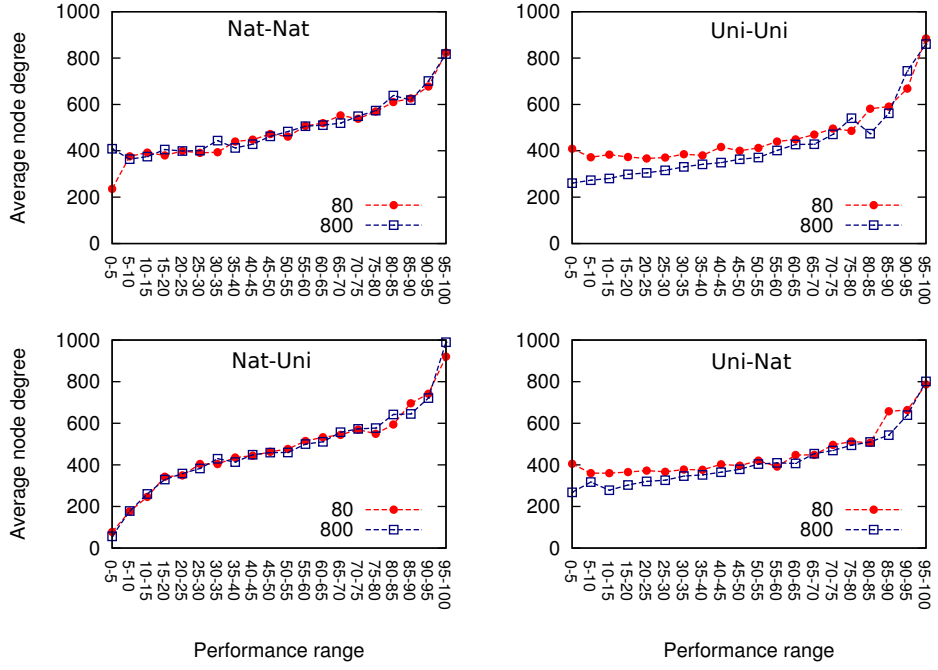

Figure B.2
Average number of incident edges on a pseudosense node in LKB vs. recall performance of UKB on the instances tagged with that specific pseudosense. We present the results for all configurations and for two sizes of the training data: 80 and 800 sentences.

trend is almost identical for the two training data sizes, namely, 80 and 800. This shows that the direct proportionality of node degree and disambiguation performance holds for different sizes of training data. Recall that the value of K—the maximum number of top-ranking related words used for LKB enrichment—was fixed (cf. Section 6.5.1). This explains why the average node degree values belonging to the two highly different sizes of the training data (i.e., 80 and 800 sentences) are comparable in Figure B.2. In fact, as the number of training sentences increases, more reliable sets of related words get selected that are likely to provide semantic edges that are more beneficial. However, the value of K, and hence the number of additional edges, remains almost constant across different sizes of the training data.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI no. 259234.

References

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies 2009: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-09)*, pages 19–27, Boulder, CO.

Agirre, Eneko, Olatz Ansa, David Martinez, and Eduard Hovy. 2001. Enriching WordNet concepts with topic signatures. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, pages 23–28, Pittsburg, PA.

Agirre, Eneko and Oier Lopez de Lacalle. 2004. Publicly available topic signatures for all WordNet nominal senses. In *Proceedings of the LREC*, pages 1,123–1,126, Lisbon.

Agirre, Eneko, Oier Lopez de Lacalle, and Aitor Soroa. 2009. Knowledge-based WSD on specific domains: Performing better than generic supervised WSD. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1,501–1,506, Pasadena, CA.

Agirre, Eneko, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based Word Sense Disambiguation. *Computational Linguistics*, 40(1):57–84.

Agirre, Eneko and David Martínez. 2004. Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of EMNLP*, pages 25–32, Barcelona.

Agirre, Eneko and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 33–41, Athens.

Banko, Michele and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse.

Bergsma, Shane, Dekang Lin, and Randy Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 59–68, Honolulu, HI.

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

Buchanan, Bruce G. and David C. Wilkins, editors. 1993. *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

Chambers, Nathanael and Dan Jurafsky. 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 '10, pages 445–453, Uppsala.

Chan, Yee Seng and Hwee Tou Ng. 2005a. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the* 20th *National Conference on Artificial Intelligence (AAAI)*, pages 1,037–1,042, Pittsburgh, PA.

Chan, Yee Seng and Hwee Tou Ng. 2005b. Word Sense Disambiguation with distribution estimation. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1,010–1,015, Edinburgh.

Chan, Yee Seng and Hwee Tou Ng. 2007. Domain adaptation with active learning for Word Sense Disambiguation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 49–56, Prague.

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

Cuadros, Montse and German Rigau. 2006. Quality assessment of large scale knowledge resources. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 534–541, Sydney.

Cuadros, Montse and German Rigau. 2007. SemEval-2007 task 16: Evaluation of wide coverage knowledge resources. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 81–86, Prague.

Cuadros, Montse and German Rigau. 2008. KnowNet: Building a large net of knowledge from the Web. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 161–168, Manchester.

Curran, James R. and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the Tenth Conference of the European Chapter of the Association for Computational Linguistics - Volume 1*, pages 91–98, Budapest.

Di Marco, Antonio and Roberto Navigli. 2013. Clustering and diversifying Web search results with graph-based Word Sense Induction. *Computational Linguistics*, 39(3):709–754.

Erk, Katrin. 2007. A simple, similarity-based model for Selectional Preferences. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 216–223, Prague.

Erk, Katrin, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse Selectional Preferences. *Computational Linguistics*, 36(4):723–763.

Escudero, Gerard, Lluís Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised Word Sense Disambiguation systems. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 172–180, Hong Kong.

Faralli, Stefano and Roberto Navigli. 2012. A new minimally-supervised framework for Domain Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1,411–1,422, Jeju.

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

Flati, Tiziano and Roberto Navigli. 2012. The CQC algorithm: Cycling in graphs to semantically enrich and enhance a bilingual dictionary. *Journal of Artificial Intelligence Research (JAIR)*, 43:135–171.

Gale, William, Kenneth Church, and David Yarowsky. 1992a. Work on statistical methods for Word Sense Disambiguation. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60, Cambridge, MA.

Gale, William A., Kenneth Church, and David Yarowsky. 1992b. A method for disambiguating word senses in a corpus. *Computers and the Humanities*, 26:415–439.

Gaustad, Tanja. 2001. Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Proceedings of the Student Research Workshop of the 39th Annual Meeting of the Association for Computational Linguistics (ACL/EACL 2001)*, pages 61–66, Toulouse.

Graff, David and Christopher Cieri. 2003. English gigaword, LDC2003T05. In *Linguistic Data Consortium*. Philadelphia, PA.

Haveliwala, Taher H. 2002. Topic-sensitive PageRank. In *Proceedings of the* 11th *International Conference on World Wide Web (WWW 2002)*, pages 517–526, Honolulu, HI.

Hovy, Eduard H., Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27.

Hughes, Thad and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 581–589, Prague.

Ide, Nancy, Collin F. Baker, Christiane Fellbaum, and Rebecca J. Passonneau. 2010. The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 68–73, Uppsala.

Jurgens, David and Keith Stevens. 2011. Measuring the impact of sense similarity on Word Sense Induction. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, EMNLP '11, pages 113–123, Edinburgh.

Khapra, Mitesh, Anup Kulkarni, Saurabh Sohoney, and Pushpak Bhattacharyya. 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)*, pages 1,532–1,541, Uppsala.

Kilgarriff, Adam and Joseph Rosenzweig. 2000. English Senseval: Report and results. In *Proceedings of the 2nd Conference on Language Resources and Evaluation (LREC)*, pages 1,239–1,244, Athens.

Leacock, Claudia, Martin Chodorow, and George Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–166.

Lee, Yoong Keok and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for Word Sense Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '02, pages 41–48, Philadelphia, PA.

Lin, Chin-Yew and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 495–501, Saarbrücken.

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

Litkowski, Ken. 2004. Senseval-3 task: Word Sense Disambiguation of WordNet glosses. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 13–16, Barcelona.

Lu, Zhimao, Haifeng Wang, Jianmin Yao, Ting Liu, and Sheng Li. 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*, pages 457–464, Sydney.

Marcus, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*, pages 114–119, Plainsboro, NJ.

Martinez, David. 2004. *Supervised Word Sense Disambiguation: Facing Current Challenges*. Ph.D. Thesis. University of the Basque Country, Spain.

Martinez, David, Oier Lopez de Lacalle, and Eneko Agirre. 2008. On the use of automatically acquired examples for all-nouns Word Sense Disambiguation. *Journal of Artificial Intelligence Research*, 33(1):79–107.

Matuschek, Michael and Iryna Gurevych. 2013. Dijkstra-WSA: A graph-based approach to word sense alignment. *Transactions of the Association for Computational Linguistics (TACL)*, (1):151–164.

McCarthy, Diana, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 279–286, Barcelona.

Mihalcea, Rada. 2002. Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Language Resources and Evaluations (LREC)*, pages 1,407–1,411, Las Palmas.

Mihalcea, Rada. 2007. Using Wikipedia for automatic Word Sense Disambiguation. In *Proceedings of NAACL-HLT-07*, pages 196–203, Rochester, NY.

Mihalcea, Rada, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona.

Mihalcea, Rada and Dan Moldovan. 1999. An automatic method for generating sense tagged corpora. In *Proceedings AAAI '99*, pages 461–466, Orlando, FL.

Mihalcea, Rada and Dan Moldovan. 2001. eXtended WordNet: Progress report. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, pages 95–100, Pittsburgh, PA.

Miller, George A., R. T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: An online lexical database. *International Journal of Lexicography*, 3(4):235–244.

Miller, George A., Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, NJ.

Moro, Andrea, Roberto Navigli, Francesco Maria Tucci, and Rebecca J. Passonneau. 2014. Annotating the MASC Corpus with BabelNet. In *Proceedings of LREC*, pages 4,214–4,219, Reykjavic.

Moro, Andrea, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Nakov, Preslav I. and Marti A. Hearst. 2003. Category-based pseudowords. In *HLT-NAACL 2003–Short Papers*, pages 67–69, Edmonton.

Navigli, Roberto. 2005. Semi-automatic extension of large-scale linguistic knowledge bases. In *Proceedings of FLAIRS-05*, pages 548–553, Clearwater Beach, FL.

Navigli, Roberto. 2008. A structural approach to the automatic adjudication of word sense disagreements. *Journal of Natural Language Engineering*, 14(4):293–310.

Navigli, Roberto. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Navigli, Roberto. 2012. A quick tour of word sense disambiguation, induction and related approaches. In *Proceedings of the 38th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 115–129, Spindleruv Mlyn.

Navigli, Roberto and Mirella Lapata. 2010. An experimental study on graph connectivity for unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Anaylsis and Machine Intelligence*, 32(4):678–692.

Navigli, Roberto and Simone Paolo Ponzetto. 2012a. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Navigli, Roberto and Simone Paolo Ponzetto. 2012b. Joining forces pays off: Multilingual joint Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1,399–1,410, Jeju.

Navigli, Roberto and Daniele Vannella. 2013. SemEval-2013 task 11: Evaluating Word Sense Induction and Disambiguation within an end-user application. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013)*, pages 193–201, Atlanta, GA.

Otrusina, Lubomir and Pavel Smrz. 2010. A new approach to pseudoword generation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 1,195–1,199, Valletta.

Palmer, Martha, Hoa Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163.

Pham, Thanh Phong, Hwee Tou Ng, and Wee Sun Lee. 2005. Word Sense Disambiguation with semi-supervised learning. In *Proceedings of the 20th National Conference on Artificial Intelligence*, AAAI'05, pages 1,093–1,098, Pittsburgh, PA.

Pilehvar, Mohammad Taher, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1,341–1,351, Sofia.

Pilehvar, Mohammad Taher and Roberto Navigli. 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 2013)*, pages 1,100–1,109, Atlanta, GA.

Ponzetto, Simone Paolo and Roberto Navigli. 2010. Knowledge-rich Word Sense Disambiguation rivaling supervised system. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1,522–1,531, Uppsala.

Pradhan, Sameer, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007a. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague.

Pradhan, Sameer S., Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007b.

OntoNotes: A unified relational semantic representation. In *Proceedings of the 1st International Conference on Semantic Computing (ICSC)*, pages 517–526, Irvine, CA.

Sanderson, Mark and C. J. Van Rijsbergen. 1999. The impact on retrieval effectiveness of skewed frequency distributions. *ACM Transactions on Information Systems*, 17:440–465.

Schütze, Hinrich. 1992. Dimensions of meaning. In *Supercomputing '92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 787–796, Los Alamitos, CA.

Shen, Hui, Razvan Bunescu, and Rada Mihalcea. 2013. Coarse to fine grained sense disambiguation in Wikipedia. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 22–31, Atlanta, GA.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, HI.

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

Vannella, Daniele, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli. 2014. Validating and extending semantic knowledge bases using video games with a purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1,294–1,304, Baltimore, MD.

Venhuizen, Noortje J., Valerio Basile, Kilian Evang, and Johan Bos. 2013. Gamification for word sense labeling. In *Proceedings of the International Conference on Computational Semantics (IWCS)*, pages 397–403, Potsdam.

Wang, Xinglong and John Carroll. 2005. Word Sense Disambiguation using sense examples automatically acquired from a second language. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 547–554, Vancouver.

Yarowsky, David. 1993. One sense per collocation. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 266–271, Princeton, NJ.

Yarowsky, David. 1995. Unsupervised Word Sense Disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA.

Zhong, Zhi and Hwee Tou Ng. 2009. Word Sense Disambiguation for all words without hard labor. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1,616–1,622, Pasadena, CA.

Zhong, Zhi and Hwee Tou Ng. 2010. It makes sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 78–83, Uppsala.

Zipf, George K. 1949. *Human Behavior and the Principle of Least-Effort*. Addison-Wesley, Cambridge, MA.