Random Walks for Knowledge-Based Word Sense Disambiguation

Eneko Agirre*
IXA NLP group
University of the Basque Country

Oier López de Lacalle**
University of Edinburgh
IKERBASQUE
Basque Foundation for Science

Aitor Soroa†
IXA NLP group
University of the Basque Country

Word Sense Disambiguation (WSD) systems automatically choose the intended meaning of a word in context. In this article we present a WSD algorithm based on random walks over large Lexical Knowledge Bases (LKB). We show that our algorithm performs better than other graph-based methods when run on a graph built from WordNet and eXtended WordNet. Our algorithm and LKB combination compares favorably to other knowledge-based approaches in the literature that use similar knowledge on a variety of English data sets and a data set on Spanish. We include a detailed analysis of the factors that affect the algorithm. The algorithm and the LKBs used are publicly available, and the results easily reproducible.

1. Introduction

Word Sense Disambiguation (WSD) is a key enabling technology that automatically chooses the intended sense of a word in context. It has been the focus of intensive research since the beginning of Natural Language Processing (NLP), and more recently it has been shown to be useful in several tasks such as parsing (Agirre, Baldwin, and Martinez 2008; Agirre et al. 2011), machine translation (Carpuat and Wu 2007; Chan, Ng, and Chiang 2007), information retrieval (Pérez-Agüera and Zaragoza 2008; Zhong and Ng 2012), question answering (Surdeanu, Ciaramita, and Zaragoza 2008), and

* Informatika Fakultatea, Manuel Lardizabal 1, 20018 Donostia, Basque Country. E-mail: e.agirre@ehu.es.
** IKERBASQUE, Basque Foundation for Science, 48011, Bilbao, Basque Country.
E-mail: oier.lopezdelacalle@gmail.com.
† Informatika Fakultatea, Manuel Lardizabal 1, 20018 Donostia, Basque Country. E-mail: a.soroa@ehu.es.

Submission received: 30 July 2011; Revised submission received: 15 November 2012; Accepted for publication: 17 February 2013.

doi:10.1162/COLI_a_00164

summarization (Barzilay and Elhadad 1997). WSD is considered to be a key step in order to approach language understanding beyond keyword matching.

The best performing WSD systems are currently those based on supervised learning, as attested in public evaluation exercises (Snyder and Palmer 2004; Pradhan et al. 2007), but they need large amounts of hand-tagged data, which is typically very expensive to produce. Contrary to lexical-sample exercises (where plenty of training and testing examples for a handful of words are provided), all-words exercises (which comprise all words occurring in a running text, and where training data is more scarce) show that only a few systems beat the most frequent sense (MFS) heuristic, with small differences. For instance, the best system in SensEval-3 scored 65.2 F1, compared to 62.4 (Snyder and Palmer 2004). The best current state-of-the-art WSD system (Zhong and Ng 2010), outperforms the MFS heuristic by 5% to 8% in absolute F1 scores on the SensEval and SemEval fine-grained English all words tasks.

The causes of the small improvement over the MFS heuristic can be found in the relatively small amount of training data available (**sparseness**) and the problems that arise when the supervised systems are applied to different corpora from that used to train the system (**corpus mismatch**) (Ng 1997; Escudero, Márquez, and Rigau 2000). Note that most of the supervised systems for English are trained over SemCor (Miller et al. 1993), a half-a-million word subset of the Brown Corpus made available from the WordNet team, and DSO (Ng and Lee 1996), comprising 192,800 word occurrences from the Brown and WSJ corpora corresponding to the 191 most frequent nouns and verbs. Several researchers have explored solutions to sparseness. For instance, Chan and Ng (2005) present an unsupervised method to obtain training examples from bilingual data, which was used together with SemCor and DSO to train one of the best performing supervised systems to date (Zhong and Ng 2010).

In view of the problems of supervised systems, knowledge-based WSD is emerging as a powerful alternative. Knowledge-based WSD systems exploit the information in a lexical knowledge base (LKB) to perform WSD. They currently perform below supervised systems on general domain data, but are attaining performance close or above MFS without access to hand-tagged data (Ponzetto and Navigli 2010). In this sense, they provide a complementary strand of research which could be combined with supervised methods, as shown for instance in Navigli (2008). In addition, Agirre, López de Lacalle, and Soroa (2009) show that knowledge-based WSD systems can outperform supervised systems in a domain-specific data set, where MFS from general domains also fails. In this article, we will focus our attention on knowledge-based methods.

Early work for knowledge-based WSD was based on measures of similarity between pairs of concepts. In order to maximize pairwise similarity for a sequence of n words where each has up to k senses, the algorithms had to consider up to k^n sense sequences. Greedy methods were often used to avoid the combinatorial explosion (Patwardhan, Banerjee, and Pedersen 2003). As an alternative, graph-based methods are able to exploit the structural properties of the graph underlying a particular LKB. These methods are able to consider all possible combinations of occurring senses on a particular context, and thus offer a way to analyze efficiently the inter-relations among them, gaining much attention in the NLP community (Mihalcea 2005; Navigli and Lapata 2007; Sinha and Mihalcea 2007; Agirre and Soroa 2008; Navigli and Lapata 2010). The nodes in the graph represent the concepts (word senses) in the LKB, and edges in the graph represent relations between them, such as subclass and part-of. Network analysis techniques based on random walks like PageRank (Brin and Page 1998) can then be used to choose the senses that are most relevant in the graph, and thus output those senses.

In order to deal with large knowledge bases containing more than 100,000 concepts (Fellbaum 1998), previous algorithms had to extract subsets of the LKB (Navigli and Lapata 2007, 2010) or construct ad hoc graphs for each context to be disambiguated (Mihalcea 2005; Sinha and Mihalcea 2007). An additional reason for the use of custom-built subsets of ad hoc graphs for each context is that if we were using a centrality algorithm like PageRank over the whole graph, it would choose the most important senses in the LKB regardless of context, limiting the applicability of the algorithm. For instance, the word *coach* is ambiguous at least between the "sports coach" and the "transport service" meanings, as shown in the following examples:

(1) *Nadal is sharing a house with his uncle and* **coach**, *Toni, and his physical trainer, Rafael Maymo.*

(2) *Our fleet comprises* **coaches** *from 35 to 58 seats.*

If we were to run a centrality algorithm over the whole LKB, with no context, then we would always assign *coach* to the same concept, and we would thus fail to correctly disambiguate either one of the given examples.

The contributions of this article are the following: (1) A WSD method based on random walks over large LKBs. The algorithm outperforms other graph-based algorithms when using a LKB built from WordNet and eXtended WordNet. The algorithm and LKB combination compares favorably to the state-of-the-art in knowledge-based WSD on a wide variety of data sets, including four English and one Spanish data set. (2) A detailed analysis of the factors that affect the algorithm. (3) The algorithm together with the corresponding graphs are publicly available[1] and can be applied easily to sense inventories and knowledge bases different from WordNet.

The algorithm for WSD was first presented in Agirre and Soroa (2009). In this article, we present further evaluation on two more recent data sets, analyze the parameters and options of the system, compare it to the state of the art, and discuss the relation of our algorithm with PageRank and the MFS heuristic.

2. Related Work

Traditional knowledge-based WSD systems assign a sense to an ambiguous word by comparing each of its senses with those of the surrounding context. Typically, some semantic similarity metric is used for calculating the relatedness among senses (Lesk 1986; Patwardhan, Banerjee, and Pedersen 2003). The metric varies between counting word overlaps between definitions of the words (Lesk 1986) to finding distances between concepts following the structure of the LKB (Patwardhan, Banerjee, and Pedersen 2003). Usually the distances are calculated using only hierarchical relations on the LKB (Sussna 1993; Agirre and Rigau 1996). Combining both intuitions, Jiang and Conrath (1997) present a metric that combines statistics from corpus and a lexical taxonomy structure. One of the major drawbacks of these approaches stems from the fact that senses are compared in a pairwise fashion and thus the number of computations grows exponentially with the number of words—that is, for a sequence of n words where each has up to k senses they need to consider up to k^n sense sequences. Although alternatives like simulated annealing (Cowie, Guthrie, and Guthrie 1992) and conceptual density

[1] http://ixa2.si.ehu.es/ukb.

(Agirre and Rigau 1996) were tried, most of the knowledge-based WSD at the time was done in a suboptimal word-by-word greedy process, namely, disambiguating words one at a time (Patwardhan, Banerjee, and Pedersen 2003). Still, some recent work on finding predominant senses in domains has applied such similarity-based techniques with success (McCarthy et al. 2007).

Recently, graph-based methods for knowledge-based WSD have gained much attention in the NLP community (Mihalcea 2005; Navigli and Velardi 2005; Navigli and Lapata 2007; Sinha and Mihalcea 2007; Agirre and Soroa 2008; Navigli and Lapata 2010). These methods use well-known graph-based techniques to find and exploit the structural properties of the graph underlying a particular LKB. Graph-based techniques consider all the sense combinations of the words occurring on a particular context at once, and thus offer a way to analyze the relations among them with respect to the whole graph. They are particularly suited for disambiguating words in the sequence, and they manage to exploit the interrelations among the senses in the given context. In this sense, they provide a principled solution to the exponential explosion problem mentioned before, with excellent performance.

Graph-based WSD is performed over a graph composed of senses (nodes) and relations between pairs of senses (edges). The relations may be of several types (lexico-semantic, cooccurrence relations, etc.) and may have some weight attached to them. All the methods reviewed in this section use some version of WordNet as a LKB. Apart from relations in WordNet, some authors have used semi-automatic and fully automatic methods to enrich WordNet with additional relations. Mihalcea and Moldovan (2001) disambiguated WordNet glosses in a resource called eXtended WordNet. The disambiguated glosses have been shown to improve results of a graph-based system (Agirre and Soroa 2008), and we have also used them in our experiments. Navigli and Velardi (2005) enriched WordNet with cooccurrence relations semi-automatically and showed that those relations are effective in a number of graph-based WSD systems (Navigli and Velardi 2005; Navigli and Lapata 2007, 2010). More recently, Cuadros and Rigau (2006, 2007, 2008) learned automatically so-called KnowNets, and showed that the new provided relations improved WSD performance when plugged into a simple vector-based WSD system. Finally, Ponzetto and Navigli (2010) have acquired relations automatically from Wikipedia, released as WordNet++, and have shown that they are beneficial in a graph-based WSD algorithm. All of these relations are publicly available with the exception of Navigli and Velardi (2005), but note that the system is available on-line.[2]

Disambiguation is typically performed by applying a ranking algorithm over the graph, and then assigning the concepts with highest rank to the corresponding words. Given the computational cost of using large graphs like WordNet, most researchers use smaller subgraphs built on-line for each target context. The main idea of the subgraph method is to extract the subgraph whose vertices and relations are particularly relevant for the set of senses from a given input context. The subgraph is then analyzed and the most relevant vertices are chosen as the correct senses of the words.

The TextRank algorithm for WSD (Mihalcea 2005) creates a complete weighted graph (e.g., a graph in which every pair of distinct vertices is connected by a weighted edge) formed by the synsets of the words in the input context. The weight of the links joining two synsets is calculated by executing Lesk's algorithm (Lesk 1986) between them—that is, by calculating the overlap between the words in the glosses of the

[2] http://lcl.uniroma1.it/ssi.

corresponding senses. Once the complete graph is built, a random walk algorithm (PageRank) is executed over it and words are assigned to the most relevant synset. In this sense, PageRank is used as an alternative to simulated annealing to find the optimal pairwise combinations. This work is extended in Sinha and Mihalcea (2007), using a collection of semantic similarity measures when assigning a weight to the links across synsets. They also compare different graph-based centrality algorithms to rank the vertices of the complete graph. They use different similarity metrics for different POS types and a voting scheme among the centrality algorithm ranks.

In Navigli and Velardi (2005), the authors develop a knowledge-based WSD method based on lexical chains called structural semantic interconnections (SSI). Although the system was first designed to find the meaning of the words in WordNet glosses, the authors also apply the method for labeling each word in a text sequence. Given a text sequence, SSI first identifies monosemous words and assigns the corresponding synset to them. Then, it iteratively disambiguates the rest of the terms by selecting the senses that get the strongest interconnection with the synsets selected so far. The interconnection is calculated by searching for paths on the LKB, constrained by some hand-made rules of possible semantic patterns.

In Navigli and Lapata (2007, 2010), the authors perform a two-stage process for WSD. Given an input context, the method first explores the whole LKB in order to find a subgraph that is particularly relevant for the words of the context. The subgraph is calculated by applying a depth-first search algorithm over the LKB graph for every word sense occurring in a context. Then, they study different graph-based centrality algorithms for deciding the relevance of the nodes on the subgraph. As a result, every word of the context is attached to the highest ranking concept among its possible senses. The best results were obtained by a simple algorithm like choosing the concept for each word with the largest degree (number of edges) and by PageRank (Brin and Page 1998). We reimplemented their best methods in order to compare our algorithm with theirs on the same setting (cf. Section 6.3). In later work (Ponzetto and Navigli 2010) the authors apply a subset of their methods to an enriched WordNet with additional relations from Wikipedia, improving their results for nouns.

Tsatsaronis, Vazirgiannis, and Androutsopoulos (2007) and Agirre and Soroa (2008) also use such a two-stage process. They build the graph as before, but using breadth-first search. The first authors apply a spreading activation algorithm over the subgraph for node ranking, while the second use PageRank. In later work (Tsatsaronis, Varlamis, and Nørvåg 2010) spreading activation is compared with PageRank and other centrality measures like HITS (Kleinberg 1998), obtaining better results than in their previous work.

This work departs from earlier work in its use of the full graph, and its ability to infuse context information when computing the importance of nodes in the graph. For this, we resort to an extension of the PageRank algorithm (Brin and Page 1998), called Personalized PageRank (Haveliwala 2002), which tries to bias PageRank using a set of representative topics and thus capture more accurately the notion of importance with respect to a particular topic. In our case, we initialize the random walk with the words in the context of the target word, and thus we obtain a context-dependent PageRank. We will show that this method is indeed effective for WSD. Note that in order to use other centrality algorithms (e.g., HITS [Kleinberg 1998]), previous authors had to build a subgraph first. In principle, those algorithms could be made context-dependent when using the full graph and altering their formulae, but we are not aware of such variations.

Random walks over WordNet using Personalized PageRank have been also used to measure semantic similarity between two words (Hughes and Ramage 2007; Agirre et al. 2009). In those papers, the random walks are initialized with a single word, whereas we use all content words in the context. The results obtained by the authors, especially in the latter paper, are well above other WordNet-based methods.

Most previous work on knowledge-based WSD has presented results on one or two general domain corpora for English. We present our results on four general domain data sets for English and a Spanish data set (Màrquez et al. 2007). Alternatively, some researchers have applied knowledge-based WSD to specific domains, using different methods to adapt the method to the particular test domain. In Agirre, López de Lacalle, and Soroa (2009) and Navigli et al. (2011), the authors apply our Personalized PageRank method to a domain-specific corpus with good results. Ponzetto and Navigli (2010) also apply graph-based algorithms to the same domain-specific corpus.

3. WordNet

Most WSD work uses WordNet as the sense inventory of choice. WordNet (Fellbaum 1998) is a freely available[3] lexical database of English, which groups nouns, verbs, adjectives, and adverbs into sets of synonyms, each expressing a distinct concept (called *synset* in WordNet parlance). For instance, *coach* has five nominal senses and two verbal senses, which correspond to the following synsets:

<**coach#n1**, *manager#n2, handler#n3*>
<**coach#n2**, *private instructor#n1, tutor#n1*>
<**coach#n3**, *passenger car#n1, carriage#n1*>
<**coach#n4**, *four-in-hand#n2, coach-and-four#n1*>
<**coach#n5**, *bus#n1, autobus#n1, charabanc#n1,double-decker#n1,jitney#n1* ... >
<**coach#v1**, *train#v7*>
<**coach#v2**>

In these synsets *coach#n1* corresponds to the first nominal sense of *coach*, *coach#v1* corresponds to the first verbal sense, and so on. Each of the senses of coach corresponds to a different synset, and each synset contains several words with different sense numbers. For instance, the first nominal sense of *coach* has two synonyms: *manager* in its second sense and *handler* in its third sense. As a synset can be identified by any of its words in a particular sense number, we will use a word and sense number to represent the full concept. Each synset has a descriptive gloss (e.g., *a carriage pulled by four horses with one driver* for *coach#n4*, or *drive a coach* for *coach#v2*). The examples correspond to the current version of WordNet (3.1), but the sense differences have varied across different versions. There exist automatic mappings across versions (Daude, Padro, and Rigau 2000), but they contain small errors. In this article we will focus on WordNet versions 1.7 and 2.1, which have been used to tag the evaluation data sets used in this article (cf. Section 6).

The synsets in WordNet are interlinked with conceptual-semantic and lexical relations. Examples of conceptual-semantic relations are hypernymy, which corresponds to the superclass or is-a relation, and holonymy, the part-of relation. Figure 1 shows two small regions of the graph around three synsets of the word *coach*, including several conceptual-semantic relations and lexical relations. For example, the figure shows that concept *trainer#n1* is a *coach#n1* (hypernymy relation), and that *seat#n1* is a part of *coach#n5* (holonymy relation). The figure only shows a small subset of the relations

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

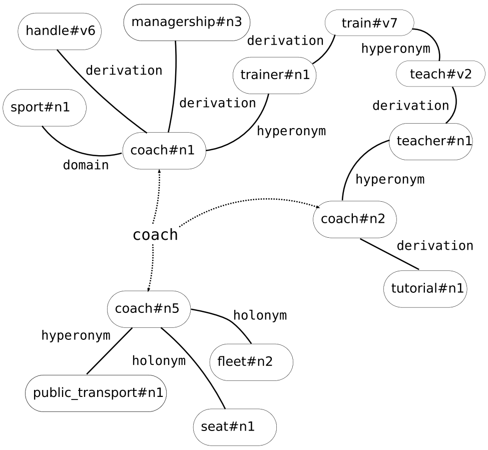

Figure 1
Example showing three senses of coach, with links to related concepts.

for three synsets of coach. If we were to show the relations of the rest of the synsets in WordNet we would end up with a densely connected graph, where one can go from one synset to another following the semantic relations. In addition to purely conceptual-semantic relations which hold between synsets, there are also lexical relations which hold between specific senses. For instance, *angry#a2* is the antonym of *calm#a2* and a derivation relation exists between *handler#n3* and *handle#v6*, meaning that *handler* is a derived form of *handle* and that the third nominal sense of *handler* is related to the sixth verbal sense of *handle*. Although lexical relations hold only between two senses, we generalize to the whole synset. This generalization captures the notion that if *handler#n3* is related by derivation to *handle#v6*, then *coach#n1* is also semantically related to *handle#v6* (as shown in Figure 1).

In addition to these relations, we also use the relation between each synset and the words in the glosses. Most of the words in the glosses have been manually associated with their corresponding senses, and we can thus produce a link between the synset being glossed, and the synsets of each of the words in the gloss. For instance, following one of the given glosses, a gloss relation would be added between *coach#v2* and *drive#v2*. The gloss relations were not available prior to WordNet 3.0, and we thus used automatically disambiguated glosses for WordNet 1.7 and WordNet 2.1, as made freely available in the eXtended WordNet (Mihalcea and Moldovan 2001). Note also that the eXtended WordNet provided about 550,000 relations, whereas the disambiguated glosses made available with WordNet 3.0 provide around 339,000 relations. We compare the performance of XWN relations and WordNet 3.0 gloss relations in Section 6.4.4.

Table 1 summarizes the most relevant relations (with less frequent relations grouped as "other"). The table also lists how we grouped the relations, and the overall counts. Note that inverse relations are not counted, as their numbers equal those of the original relation. In Section 6.4.5 we report the impact of the relations in the behavior of the system. Overall, the graph for WordNet 1.7 has $109,359$ vertices (concepts) and $620,396$ edges (relations between concepts). Note that there is some overlap between XWN and other types of relations. For instance, the hypernym of *coach#n4* is *carriage#n2*, which is also present in its gloss. Note that most of the relation types relate concepts from the same part of speech, with the exception of derivation and XWN.

Table 1
Relations and their inverses in WordNet 1.7, how we grouped them, and overall counts. XWN refers to relations from the disambiguated glosses in eXtended WordNet.

relation	inverse	group	counts
hypernymy	hyponymy	TAX	89,078
derivation	derivation	REL	28,866
holonymy	meronymy	MER	21,260
antonymy	antonymy	ANT	7,558
other	other	REL	3,134
xwn	xwn^{-1}	XWN	551,551

Finally, we have also used the Spanish WordNet (Atserias, Rigau, and Villarejo 2004). In addition to the native relations, we also added relations from the eXtended WordNet. All in all, it contains $105,501$ vertices and $623,316$ relations.

3.1 Representing WordNet as a Graph

An LKB such as WordNet can be seen as a set of concepts and relations among them, plus a dictionary, which contains the list of words (typically word lemmas) linked to the corresponding concepts (senses). WordNet can be thus represented as a graph $G = (V, E)$. V is the set of nodes, where each node represents one concept ($v_i \in V$), and E is the set of edges. Each relation between concepts v_i and v_j is represented by an edge $e_{i,j} \in E$. We ignore the relation type of the edges. If two WordNet relations exist between two nodes, we only represent one edge, and ignore the type of the relation. We chose to use undirected relations between concepts, because most of the relations are symmetric and have their inverse counterpart (cf. Section 3), and in preliminary work we failed to see any effect using directed relations.

In addition, we also add vertices for the dictionary words, which are linked to their corresponding concepts by directed edges (cf. Figure 1). Note that monosemous words will be related to just one concept, whereas polysemous words may be attached to several. Section 5.2 explains the reason for using directed edges, and also mentions an alternative to avoid introducing these vertices.

4. PageRank and Personalized PageRank

The PageRank random walk algorithm (Brin and Page 1998) is a method for ranking the vertices in a graph according to their relative structural importance. The main idea of PageRank is that whenever a link from v_i to v_j exists in a graph, a vote from node i to node j is produced, and hence the rank of node j increases. In addition, the strength of the vote from i to j also depends on the rank of node i: The more important node i is, the more strength its votes will have. Alternatively, PageRank can also be viewed as the result of a random walk process, where the final rank of node i represents the probability of a random walk over the graph ending on node i, at a sufficiently large time.

Let G be a graph with N vertices $v_1, \ldots, v_N$ and d_i be the outdegree of node i; let M be a $N \times N$ transition probability matrix, where $M_{ji} = \frac{1}{d_i}$ if a link from i to j exists, and zero otherwise. Then, the calculation of the **PageRank Vector P** over G is equivalent to resolving Equation (1).

$$\mathbf{P} = cM\mathbf{P} + (1 - c)\mathbf{v} \tag{1}$$

In the equation, $\mathbf{v}$ is a $N \times 1$ stochastic vector and c is the so-called **damping factor**, a scalar value between 0 and 1. The first term of the sum on the equation models the voting scheme described in the beginning of the section. The second term represents, loosely speaking, the probability of a surfer randomly jumping to any node (e.g., without following any paths on the graph). The damping factor, usually set in the [0.85..0.95] range, models the way in which these two terms are combined at each step.

The second term in Equation (1) can also be seen as a smoothing factor that makes any graph fulfill the property of being aperiodic and irreducible, and thus guarantees that the PageRank calculation converges to a unique stationary distribution.

In the traditional PageRank formulation the vector $\mathbf{v}$ is a stochastic normalized vector whose element values are all $\frac{1}{N}$, thus assigning equal probabilities to all nodes in the graph in the case of random jumps. However, as pointed out by Haveliwala (2002), the vector $\mathbf{v}$ can be non-uniform and assign stronger probabilities to certain kinds of nodes, effectively biasing the resulting PageRank vector to prefer these nodes. For example, if we concentrate all the probability mass on a unique node i, all random jumps on the walk will return to i and thus its rank will be high; moreover, the high rank of i will make all the nodes in its vicinity also receive a high rank. Thus, the importance of node i given by the initial distribution of $\mathbf{v}$ spreads along the graph on successive iterations of the algorithm. As a consequence, the $\mathbf{P}$ vector can be seen as representing the relevance of every node in the graph *from the perspective of node i*.

In this article, we will use **Static PageRank** to refer to the case when a uniform $\mathbf{v}$ vector is used in Equation (1); and whenever a modified $\mathbf{v}$ is used, we will call it **Personalized PageRank**. The next section shows how we define a modified $\mathbf{v}$.

PageRank is actually calculated by applying an iterative algorithm that computes Equation (1) successively until convergence below a given threshold is achieved, or until a fixed number of iterations are executed. Following usual practice, we used a damping value of 0.85 and finish the calculations after 30 iterations (Haveliwala 2002; Langville and Meyer 2003; Mihalcea 2005). Some preliminary experiments with higher iteration counts showed that although sometimes the node ranks varied, the relative order among particular word synsets remained stable after the initial iterations (cf. Section 6.4 for further details). Note that, in order to discard the effect of **dangling nodes** (i.e., nodes without outlinks) one would need to slightly modify Equation (1) following Langville and Meyer (2003).[4] This modification is not necessary for WordNet, as it does not have dangling nodes.

5. Random Walks for WSD

We tested two different methods to apply random walks to WSD.

[4] The equation becomes $\mathbf{P} = cM\mathbf{P} + (c\mathbf{a} + (1 - c)\mathbf{e})\mathbf{v}$, where $a_i = 1$ if node i is a dangling node, and 0 otherwise, and $\mathbf{e}$ is a vector of all ones.

5.1 Static PageRank, No Context

If we apply traditional PageRank over the whole WordNet, we get a context-independent ranking of word senses. All concepts in WordNet get ranked according to their PageRank value. Given a target word, it suffices to check which is the relative ranking of its senses, and the WSD system would output the one ranking highest. We call this application of PageRank to WSD Static PageRank STATIC for short, as it does not change with the context, and we use it as a baseline.

As the PageRank measure over undirected graphs for a node is closely related to the degree of the node, the Static PageRank returns the most predominant sense according to the number of relations the senses have. We think that this is closely related to the Most Frequent Sense attested in general corpora, as the lexicon builders would tend to assign more relations to the most predominant sense. In fact, our results (cf. Section 6.4.5) show that this is indeed the case for the English WordNet.

5.2 Personalized PageRank, Using Context

Static PageRank is independent of context, but this is not what we want in a WSD system. Given an input piece of text we want to disambiguate all content words in the input according to the relationships among them. For this we can use Personalized PageRank (PPR for short) over the whole WordNet graph.

Given an input text (e.g., a sentence), we extract the list $W_i \; i = 1 \ldots m$ of content words (i.e., nouns, verbs, adjectives, and adverbs) that have an entry in the dictionary, and thus can be related to LKB concepts. As a result of the disambiguation process, every LKB concept receives a score. Then, for each target word to be disambiguated, we just choose its associated concept in G with maximum score.

In order to apply Personalized PageRank over the LKB graph, the context words are first inserted into the graph G as nodes, and linked with directed edges to their respective concepts. Then, the Personalized PageRank of the graph G is computed by concentrating the initial probability mass uniformly over the newly introduced word nodes. As the words are linked to the concepts by directed edges, they act as source nodes injecting mass into the concepts they are associated with, which thus become relevant nodes, and spread their mass over the LKB graph. Therefore, the resulting Personalized PageRank vector can be seen as a measure of the structural relevance of LKB concepts in the presence of the input context.

Making the edges from words to concepts directed is important, as the use of undirected edges will move part of the probability mass in the concepts to the word nodes. Note the contrast with the edges representing relations between concepts, which are undirected (cf. Section 3.1).

Alternatively, we could do without the word nodes, concentrating the initial probability mass on the senses of the words under consideration. Such an initialization over the graph with undirected edges between synset nodes is equivalent to initializing the walk on the words in a graph with undirected edges between synset nodes and directed nodes from words to synsets. We experimentally checked that the results of both alternatives are indistinguishable. Although the alternative without nodes is marginally more efficient, we keep the word nodes as they provide a more intuitive and appealing formalization.

One problem with Personalized PageRank is that if one of the target words has two senses that are related by semantic relations, those senses reinforce each other, and could thus dampen the effect of the other senses in the context. Although one could remove direct edges between competing senses from the graph, it is quite rare that those senses are directly linked, and usually a path with several edges is involved. With this observation in mind we devised a variant called **word-to-word heuristic** (PPR$_{w2w}$ for short), where we run Personalized PageRank separately for each target word in the context, that is, for each target word W_i, we concentrate the initial probability mass in the senses of the rest of the words in the context of W_i, but not in the senses of the target word itself, so that context words increase their relative importance in the graph. The main idea of this approach is to avoid biasing the initial score of concepts associated with target word W_i, and let the surrounding words decide which concept associated with W_i has more relevance. Contrary to the previous approach, PPR$_{w2w}$ does not disambiguate all target words of the context in a single run, which makes it less efficient (cf. Section 6.4).

Figure 2 illustrates the disambiguation of a sample sentence. The STATIC method (not shown in the figure) would choose the synset *coach#n1* for the word *coach* because it is related to more concepts than other senses, and because those senses are related to concepts that have a high degree (for instance, *sport#1*). The PPR method (left side of Figure 2) concentrates the initial mass on the content words in the example. After running the iterative algorithm, the system would return *coach#n1* as the result for the target word *coach*. Although the words in the sentence clearly indicate that the correct synset in this sentence corresponds to *coach#n5*, the fact that *teacher#n1* is related to *trainer#n1* in WordNet causes both *coach#n2* and *coach#n1* to reinforce each other, and make their pagerank higher. The right side of Figure 2 depicts the PPR$_{w2w}$ method, where the word *coach* is not activated. Thus, there is no reinforcement between *coach* senses, and the method would correctly choose *coach#n5* as the proper synset.

6. Evaluation

WSD literature has used several measures for evaluation. **Precision** is the percentage of correctly disambiguated instances divided by the number of instances disambiguated. Some systems don't disambiguate all instances, and thus the precision can be high even if the system disambiguates a handful of instances. In our case, when a word has

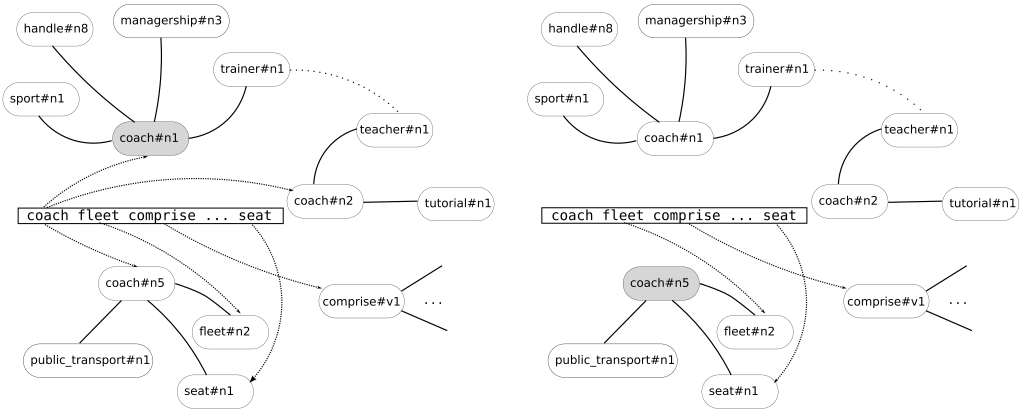

Figure 2
Portion of WordNet to illustrate the disambiguation of *coach* in the sentence *Our fleet comprises coaches from 35 to 58 seats*. Each word in the sentence (shown partially) is linked to all its synsets. The path between *trainer#n1* and *teacher#1* is omitted for brevity (see Figure 1). The left part shows the PPR method, and the right part shows the PPR$_{w2w}$ method.

two senses with the same PageRank value, our algorithm does not return anything, because it abstains from returning a sense in the case of ties. In contrast, **recall** measures the percentage of correctly disambiguated instances divided by the total number of instances to be disambiguated. This measure penalizes systems that are unable to return a solution for all instances. Finally, the harmonic mean between precision and recall (**F1**) combines both measures. F1 is our main measure of evaluation, as it provides a balanced measure between the two extremes. Note that a system that returns a solution for all instances would have equal precision, recall, and F1 measures.

In our experiments we build a context of at least 20 content words for each sentence to be disambiguated, taking the sentences immediately before and after it in the case that the original sentence was too short. The parameters for the PageRank algorithm were set to 0.85 and 30 iterations following standard practice (Haveliwala 2002; Langville and Meyer 2003; Mihalcea 2005). The post hoc impact of those and other parameters has been studied in Section 6.4.

The general domain data sets used in this work are the SensEval-2 (S2AW) (Snyder and Palmer 2004), SensEval-3 (S3AW) (Palmer et al. 2001), and SemEval-2007 fine-grained (S07AW) (Palmer et al. 2001; Snyder and Palmer 2004; Pradhan et al. 2007) and coarse grained all-words data sets (S07CG) (Navigli, Litkowski, and Hargraves 2007). All data sets have been produced similarly: A few documents were selected for tagging, at least two annotators tagged nouns, verbs, adjectives, and adverbs, inter-tagger agreement was measured, and the discrepancies between taggers were solved. The first two data sets are labeled with WordNet 1.7 tags, the third uses WordNet 2.1 tags, and the last one uses coarse-grained senses that group WordNet 2.1 senses. We run our system using WordNet 1.7 relations and senses for the first two data sets, and WordNet 2.1 for the other two. Section 6.4.3 explores the use of WordNet 3.0 and compares the performance with the use of other versions.

Regarding the coarse senses in S07CG, we used the mapping from WordNet 2.1 senses made available by the authors of the data set. In order to return coarse grained-senses, we run our algorithm on fine-grained senses, and aggregate the scores for all senses that map to the same coarse-grained sense. We finally choose the coarse-grained sense with the highest score.

The data sets used in this article contain polysemous and monosemous words, as customary; the percentage of monosemous word occurrences in the S2AW, S3AW, S07AW, and S07CG data sets are 20.7%, 16.9%, 14.4%, and 29.9%, respectively.

6.1 Results

Table 2 shows the results as F1 of our random walk WSD systems over these data sets. We detail overall results, as well as results per part of speech, and whether there is any statistical difference with respect to the best result on each column. Statistical significance is obtained using the paired bootstrap resampling method (Noreen 1989), $p < 0.01$.

The table shows that PPR$_{w2w}$ is consistently the best method in three data sets. All in all the differences are small, and in one data set STATIC obtains the best results. The differences with respect to the best system overall are always statistically significant. In fact, it is remarkable that a simple non-contextual measure like STATIC performs so well, without the need for building subgraphs or any other manipulation. Section 6.4.6 will show that in some circumstances the performance of STATIC is much lower and analyzes the reasons for this drop. Regarding the use of the word-to-word heuristic, it consistently provides slightly better results than PPR in all four data sets. An analysis of

Table 2
Results on English data sets (F1). Best results in each column in **bold**. * Statistically significant with respect to the best result in each column.

S2AW - SensEval-2 All-Words

Method	All	N	V	Adj.	Adv.
PPR	58.7*	**71.8**	35.0	58.9	69.8
PPR$_{w2w}$	**59.7**	70.3	**40.3**	**59.8**	**72.9**
STATIC	58.0*	66.5	40.2	**59.8**	72.5

S3AW - SensEval-3 All-Words

Method	All	N	V	Adj.	Adv.
PPR	57.3*	63.7	**47.5**	61.3	**96.3**
PPR$_{w2w}$	**57.9**	**65.3**	47.2	**63.6**	**96.3**
STATIC	56.5*	62.5	47.1	62.8	**96.3**

S07AW - SemEval 2007 All-Words

Method	All	N	V	Adj.	Adv.
PPR	39.7*	51.6	34.6	–	–
PPR$_{w2w}$	41.7*	**56.0**	35.3	–	–
STATIC	**43.0**	**56.0**	**37.3**	–	–

S07CG - SemEval 2007 Coarse-grained All-Words

Method	All	N	V	Adj.	Adv.
PPR	78.1*	78.3	**73.8**	**84.0**	78.4
PPR$_{w2w}$	**80.1**	**83.6**	71.1	83.1	82.3
STATIC	79.2*	81.0	72.4	82.9	**82.8**

the performance according to the POS shows that PPR$_{w2w}$ performs better particularly on nouns, but there does not seem to be a clear pattern for the rest. In the rest of the article, we will only show the overall results, omitting those for all POS, in order not to clutter the result tables.

Our algorithms do not always return an answer, and thus the precision is higher than the F1 measure. For instance, in S2AW the percentage of instances that get an answer ranges between 95.4% and 95.6% for PPR, PPR$_{w2w}$, and STATIC. The precision for PPR$_{w2w}$ in S2AW is 61.1%, the recall is 58.4%, and F1 is 59.7%. This pattern of slightly higher values for precisions, lower values for recall, and F1 in between is repeated for all data sets, POS, and data sets. The percentage of instances that get an answer for the other data sets is higher, ranging between 98.1% in S3AW and 99.9% in S07CG.

6.2 Comparison to State-of-the-Art Systems

In this section we compare our results with the WSD systems described in Section 2, as well as the top performing supervised systems at competition time and other unsupervised systems that improved on them. Note that we do not mention all unsupervised systems participating in the competitions, but we do select the top performing ones. All results in Table 3 are given as overall F1 for all Parts of Speech, but we also report F1 for nouns in the case of S07CG, where Ponz10 (Ponzetto and Navigli 2010) reported very high results, but only for nouns. Note that the systems reported here and our system might use different context sizes.

For easier reference, Table 3 uses a shorthand for each system, whereas the text in this section includes the shorthand and the full reference the first time the shorthand is used. The shorthand uses the first letters of the first author followed by the year of the paper, except for systems which participated in SensEval and SemEval, where we use their acronym. Most systems in the table have been presented in Section 2, with a few exceptions that will be presented this section.

The results in Table 3 confirm that our system (PPR$_{w2w}$) performs on the state-of-the-art of knowledge-based and unsupervised systems, with two exceptions:

(1) Nav10 (Navigli and Lapata 2010) obtained better results on S07AW. We will compare both systems in more detail below, and also include a reimplementation in the next subsection which shows that, when using the same LKB, our method obtains better results.

(2) Although not reported in the table, an unsupervised system using automatically acquired training examples from bilingual data (Chan and Ng 2005) obtained very good results on S2AW nouns (77.2 F1, compared with our 70.3 F1 in Table 2). The automatically acquired training examples are used in addition to hand-annotated data in Zhong10 (Zhong and Ng 2010), also reported in the table (see below).

We report the best unsupervised systems in S07AW and S07CG on the same row. JU-SKNSB (Naskar and Bandyopadhyay 2007) is a system based on an extended version

Table 3
Comparison with state-of-the-art results (F1). The top rows report knowledge-based and unsupervised systems, followed by our system (PPR$_{w2w}$). Below we report systems that use annotated data to some degree: (1) MFS or counts from hand-annotated corpora, (2) fully supervised systems, including the best supervised participants in each exercise. Best result among unsupervised systems in each column is shown in **bold**. Please see text for references of each system.

System	S2AW	S3AW	S07AW	S07CG	(N)
Mih05	54.2	52.2			
Sinha07	57.6	53.6			
Tsatsa10	58.8	57.4			
Agirre08		56.8			
Nav10		52.9	**43.1**		
JU-SKNSB / TKB-UO			40.2	70.2	(70.8)
Ponz10					(79.4)
PPR$_{w2w}$	**59.7**	**57.9**	41.7	**80.1**	(**83.6**)
MFS$^{(1)}$	60.1	62.3	51.4	78.9	(77.4)
IRST-DDD-00$^{(1)}$		58.3			
Nav05$^{(1)}$ / UOR-SSI$^{(1)}$		60.4		83.2	(84.1)
Best$_{sup}^{(2)}$	68.6	65.2	59.1	82.5	(82.3)
Zhong10$^{(2)}$	68.2	67.6	58.3	82.6	

of the Lesk algorithm (Lesk 1986), evaluated on S07AW. TKB-UO (Anaya-Sánchez, Pons-Porrata, and Berlanga-Llavori 2007), which was evaluated in S07CG, clusters WordNet senses and uses so-called topic signatures based on WordNet information for disambiguation. IRST-DDD-00 (Strapparava, Gliozzo, and Giuliano 2004) is a system based on WordNet domains which leverages on large unannotated corpora. They obtained excellent results, but their calculation of scores takes into account synset probabilities from SemCor, and the system can thus be considered to use some degree of supervision. We consider that systems which make use of information derived from hand-annotated corpora need to be singled out as having some degree of supervision. This includes systems using the MFS heuristic, as it is derived from hand-annotated corpora. In the case of the English WordNet, the use of the first sense also falls in this category, as the order of senses in WordNet is based on sense counts in hand-annotated corpora. Note that for wordnets in other languages, hand-annotated corpus is scarce, and thus our main results do not use this information. Section 6.4.7 analyzes the results of our system when combined with this information.

Among supervised systems, the best supervised systems at competition time are reported in a single row (Mihalcea 2002; Decadt et al. 2004; Chan, Ng, and Zhong 2007; Tratz et al. 2007). We also report Zhong10 (Zhong and Ng 2010), which is a freely available supervised system giving some of the strongest results in WSD.

We will now discuss in detail the systems that are most similar to our own. We first review the WordNet versions and relations used by each system. Mih05 (Mihalcea 2005) and Sinha07 (Sinha and Mihalcea 2007) apply several similarity methods, which use WordNet information from versions 1.7.1 and 2.0, respectively, including all relations and the text in the glosses.[5] Tsatsa10 (Tsatsaronis, Varlamis, and Nørvåg 2010) uses WordNet 2.0. Agirre08 (Agirre and Soroa 2008) experimented with several LKBs formed by combining relations from different sources and versions, including WordNet 1.7 and eXtended WordNet. Nav05 and Nav10 (Navigli and Velardi 2005; Navigli and Lapata 2010) use WordNet 2.0, enriched with manually added co-occurrence relations which are not publicly available.

We can see in Table 3 that the combination of Personalized PageRank and LKB presented in this article outperforms both Mih05 and Sinha07. In order to factor out the difference in the WordNet version, we performed experiments using WN2.1 and eXtended WordNet, yielding 58.7 and 56.5 F1 for S2AW and S3AW, respectively. Although a head-to-head comparison is not possible, the systems use similar information: Although they use glosses, our algorithm cannot directly use the glosses, and thus we use disambiguated glosses as delivered in eXtended WordNet. All in all the results suggest that analyzing the LKB structure as a graph is preferable to computing pairwise similarity measures over synsets to build a custom graph and then applying graph measures. The results of various in-house experiments replicating Mih05 also confirmed this observation. Note also that our methods are simpler than the combination strategy used in Sinha07.

Nav05 (Navigli and Velardi 2005) uses a knowledge-based WSD method based on lexical chains called structural semantic interconnections (SSI). The SSI method was evaluated on the SensEval-3 data set, as shown in row Nav05 in Table 3. Note that the method labels an instance with the MFS of the word if the algorithm produces no output for that instance, which makes comparison to our system unfair, especially given the fact that the MFS performs better than SSI. In fact, it is not possible to separate

[5] Personal communication.

the effect of SSI from that of the MFS, and we thus report it as using some degree of supervision in the table. A variant of the algorithm called UOR-SSI (Navigli, Litkowski, and Hargraves 2007) (reported in the same row) used a manually added set of 70,000 relations and obtained the best results in S07CG out-of-competition,[6] even better than the best supervised method. Reimplementing SSI is not trivial, so we did not check the performance of a variant of SSI that does not use MFS and that uses the same LKB as our method. Section 6.4.7 analyzes the results of our system when combined with MFS information.

Agirre08 (Agirre and Soroa 2008) uses breadth-first search to extract subgraphs of the WordNet graph for each context to be disambiguated, and then applies PageRank. Our better results seem to indicate that using the full graph instead of those subgraphs would perform better. In order to check whether the better results are due to differences in the information used, the next subsection presents the results of our reimplementation of the systems using the same information as our full-graph algorithm.

Tsatsa10 (Tsatsaronis, Vazirgiannis, and Androutsopoulos 2007; Tsatsaronis, Varlamis, and Nørvåg 2010) also builds the graph using breadth-first search, but weighting each type of edge differently, and using graph-based measures that take into account those weights. This is in contrast to the experiments performed in this article where edges have no weight, and is an interesting avenue for future work.

Nav10 (Navigli and Lapata 2010) first builds a subgraph of WordNet composed of paths between synsets using depth-first search and then applies a set of graph centrality algorithms. The best results are obtained using the degree of the nodes, and they present two variants, depending on how they treat ties: Either they return a sense at random, or they return the most frequent sense. For fair comparison to our system (which does not use MFS as a back-off), Table 3 reports the former variant as Nav10. This system is better than ours in one data set and worse in another. They use 60,000 relations that are not publicly available, but they do not use eXtended WordNet relations. In order to check whether the difference in performance is due to the relations used or the algorithm, the next subsection presents a reimplementation of their best graph-based algorithms using the same LKB as we do. In earlier work (Navigli and Lapata 2007) they test a similar system on S3AW, but report results only for nouns, verbs, and adjectives (F1 of 61.9, 36.1, and 62.8, respectively), all of which are below the results of our system (cf. Table 2).

In Ponz10 (Ponzetto and Navigli 2010) the authors apply the same techniques as in Nav10 to a new resource called WordNet++. They report results for nouns using degree on subgraphs for the S07CG data set, as shown in Table 3. Their F1 on nouns is 79.4, lower than our results using our LKB.

6.3 Comparison with Related Algorithms

The previous section shows that our algorithm when applied to a LKB built from WordNet and eXtended WordNet outperforms other knowledge-based systems in all cases but one system in one data set. In this section we factor out algorithm and LKB, and present the results of other graph-based methods for WSD using the same WordNet versions and relations as in the previous section. As we mentioned in Section 2, ours is the only method using the full WordNet graph. Navigli and Lapata (2010) and Ponzetto and Navigli (2010) build a custom graph based on the relations in WordNet

[6] The task was co-organized by the authors.

as follows: For each sense s_i of each word in the context, a depth-first search (DFS for short) is conducted through the WordNet graph starting in s_i until another sense s_j of a word in the context is found or maximum distance is reached. The maximum distance was set by the authors to 6. All nodes and edges between s_i and s_j, inclusive, are added to the subgraph. Graph-based measures are then used to select the output senses for each target word, with degree and PageRank yielding the best results. In closely related work, Agirre and Soroa (2008) and Tsatsaronis, Varlamis, and Nørvåg (2010) use breadth-first search (BFS) over the whole graph, and keep all paths connecting senses. Note that unlike the **dfs** approach, **bfs** does not require any threshold. The subgraphs obtained by each of these methods are slightly different.

We reimplemented both strategies, namely, DFS with threshold 6 and BFS with no threshold. Table 4 shows the overall results of degree and PageRank for both kinds of subgraphs. DFS yields slightly better results than BFS but PPR$_{w2w}$ is best in all four data sets, with statistical significance.

In addition, we run PPR and PPR$_{w2w}$ on DFS and BFS subgraphs, and obtained better results than degree and PageRank in all data sets. DFS with PPR and DFS with PPR$_{w2w}$ are best in S3AW and S07AW, respectively, although the differences with PPR$_{w2w}$ are not statistically significant. PPR$_{w2w}$ on the full graph is best in two data sets, with statistical significance.

From these results we can conclude that PPR and PPR$_{w2w}$ yield the best results also for subgraphs. Regarding the use of the full graph with respect to DFS or BFS, the performances for PPR$_{w2w}$ are very similar, but using the full graph gives a small advantage. Section 6.4.5 provides an analysis of efficiency.

6.4 Analysis of Performance Factors

The behavior of the WSD system is influenced by a set of parameters that can yield different results. In our main experiments we did not perform any parameter tuning; we just used some default values which were found to be useful according to previous work. In this section we perform a post hoc analysis of several parameters on the general performance of the system, reporting F1 on a single data set, S2AW.

Table 4
Results for subgraph methods compared with our method (F1). In the *Reference* column we mention the reference system that we reimplemented. Best results in each column in **bold**. * Statistically significant with respect to the best result in each column. ^{0} No significant difference.

	Reference	S2AW	S3AW	S07AW	S07CG
DFS$_{\text{degree}}$	Nav10, Ponz10	58.4*	56.4*	40.3*	79.4*
BFS$_{\text{degree}}$		57.9*	56.5*	39.9*	79.2*
DFS$_{\text{PageRank}}$	Nav10	58.2*	56.4*	39.9*	79.6*
BFS$_{\text{PageRank}}$	Agirre08	57.7*	56.7*	39.7*	79.4*
DFS$_{\text{PPR}}$		59.3*	**58.2**	41.4^0	78.1*
BFS$_{\text{PPR}}$		58.8^0	57.5^0	41.2^0	78.8*
DFS$_{\text{PPR}_{w2w}}$		58.7^0	58.0^0	41.2*	79.7*
BFS$_{\text{PPR}_{w2w}}$		58.1*	57.9^0	**41.9**	79.5*
PPR$_{w2w}$		**59.7**	57.9^0	41.7^0	**80.1**

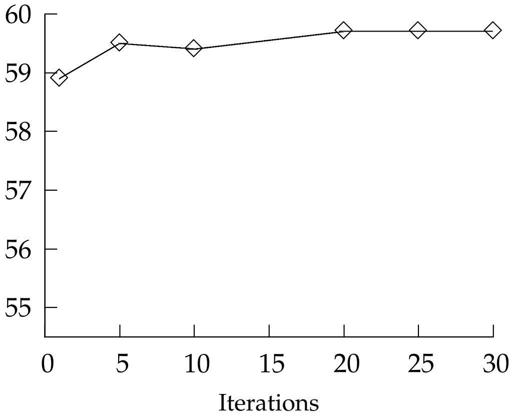

Figure 3
Convergence according to number of PageRank iterations (F1 on S2AW).

6.4.1 PageRank Parameters. The PageRank algorithm has two main parameters, the so-called damping factor and the number of iterations (or, conversely, the convergence threshold), which we set as 0.85 and 30, respectively (cf. Section 4). Figure 3 depicts the effect of varying the number of iterations. It shows that the algorithm converges very quickly: One sole iteration yields relatively high performance, and 20 iterations are enough to achieve convergence. Note also that the performance is in the [58.0, 58.5] range for iterations over 5. Note that we use the same range of F1 for the y axis of Figures 3, 4, and 5 for easier comparison.

Figure 4 shows the effect of varying the damping factor. Note that a damping factor of zero means that the PageRank value coincides with the initial probability distribution. Given the way we initialize the distribution (c.f. Section 5.2), it would mean that the algorithm is not able to disambiguate the target words. Thus, the initial value on Figure 4 corresponds to a damping factor of 0.001. On the other hand, a damping factor of 1 yields to the same results as the STATIC method (c.f. Section 5.1). The best value is attained with 0.90, with similar values around it (less than 0.5 absolute points in

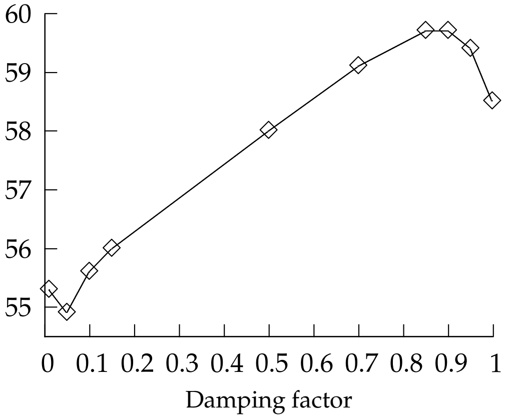

Figure 4
Varying the damping factor (F1 on S2AW).

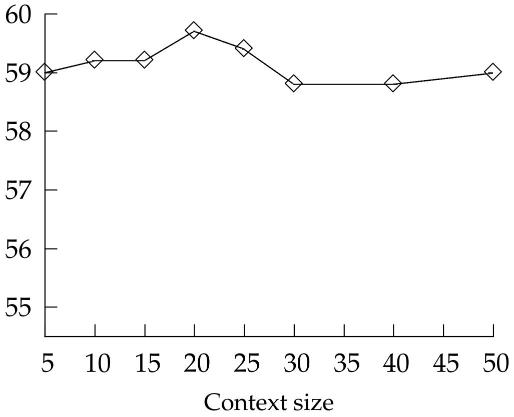

Figure 5
Varying the context size (F1 on S2AW).

variation), in agreement with previous results which preferred values in the 0.85...0.95 interval (Haveliwala 2002; Langville and Meyer 2003; Mihalcea 2005).

6.4.2 Size of Context Window. Figure 5 shows the performance of the system when trying different context windows for the target words. The best context size is for windows of 20 content words, with less than 0.5 absolute point losses for windows in the [5, 25] range.

6.4.3 Using Different WordNet Versions. There has been little research on the best strategy to use when dealing with data sets and resources attached to different WordNet versions. Table 5 shows the results for the four data sets used in this study when using different WordNet versions. Two of the data sets (S2AW and S3AW) were tagged with senses from version 1.7, S07AW with senses from version 2.1, and S07CG with coarse senses built on version 2.1 senses.

Given the fact that WordNet 3.0 is a more recent version that includes more relations, one would hope that using it would provide the best results (Cuadros and Rigau 2008; Navigli and Lapata 2010). We built a graph analogous to the ones for versions 1.7 and 2.1, but using the hand-disambiguated glosses instead of eXtended WordNet glosses. We used freely available mappings (Daude, Padro, and Rigau 2000)[7] to convert our eXtended WordNet relations to 3.0, and then the WordNet 3.0 sense results to the corresponding version. In addition, we also tested WN1.7 on S07AW and S07CG, and WN2.1 on S2AW and S3AW, also using the mappings from Daude, Padro, and Rigau (2000).

Table 5 shows that the best results are obtained using our algorithm on the same WordNet version as used in the respective data set. When testing on data sets tagged with WordNet 1.7, similar results are obtained using 2.1 or 3.0. When testing on data sets based on 2.1, 3.0 has a small lead over 1.7. In any case, the differences are small ranging from 1.4 absolute points to 0.5 points. All in all, it seems that the changes introduced

[7] `http://nlp.lsi.upc.edu/tools/download-map.php`.

Table 5
Comparing WordNet versions. Best result in each row in **bold**.

Data set	version	1.7 + xwn	2.1 + xwn	3.0 + xwn
S2AW	1.7	**59.7**	58.7	58.4
S3AW	1.7	**57.9**	56.5	56.8
S07AW	2.1	40.7	**41.7**	40.9
S07CG	2.1 coarse	79.6	**80.1**	79.6

by different versions slightly deteriorate the results, and the best strategy is to use the same WordNet version as was used for tagging.

6.4.4 Using xwn *vs. WN3.0 Gloss Relations.* WordNet 3.0 was released with an accompanying data set comprising glosses where some of the words had been manually disambiguated. In Table 6 we present the results of using these glosses with the WN3.0 graph, showing that the results are lower than using XWN relations. We also checked the use of WN3.0 gloss relation with other WordNet versions, and the results using XWN were always slightly better. We hypothesize that the better results for XWN are due to the amount of relations, with XWN holding 61% more relations than WN3.0 glosses. Still, the best relations are obtained with the combination of both kinds of gloss relations.

6.4.5 Analysis of Relations. Previous results were obtained using all the relations of WordNet and taking eXtended WordNet relations into account. In this section we analyze the effect of the relation types on the whole process, following the relation groups presented in Table 1. Table 7 shows the results when using different combinations over relation types. The eXtended WordNet XWN relations appear the most valuable when performing random walk WSD, as their performance is as good as when using the whole graph, and they produce a large drop when ablated from the graph. Ignoring antonymy relations produces a small improvement, but the differences between using all the relations, eliminating antonyms, and using XWN relations only are too small to draw any further conclusions. It seems that given the XWN relations (the most numerous, cf. Section 3.1), our algorithm is fairly robust to the addition or deletion of other kinds of relations (less numerous).

6.4.6 Behavior with Respect to STATIC *and MFS.* The high results of the very simple STATIC method (PageRank with no context) seems to imply that there is no need to use context for disambiguation. Our intuition was that the synsets which correspond to the most

Table 6
Comparing XWN and WN3.0 gloss relations, separately and combined. Best result in each row in **bold**.

Data set	3.0 + XWN	3.0 + gloss	3.0 + XWN + gloss
S2AW	58.4	58.1	**58.8**
S3AW	**56.8**	51.7	56.1
S07AW	40.9	38.8	**42.2**
S07CG	79.6	78.9	**80.2**

Table 7
Analysis of relation types. The first column shows the performance using just that relation type. The second shows the combination of TAX and each type. The last column shows all relations except the corresponding type.

relation	single	+ TAX	ablation
TAX	37.4	–	**59.9**
ANT	19.1	42.1	**59.9**
MER	23.4	36.4	59.6
REL	35.4	46.1	59.6
XWN	**59.9**	**59.8**	47.1
Reference system (all relations)			59.7

frequent senses would get more relations. We thus computed the correlation between systems, gold tags, and MFS. In order to make the correlation results comparable to the figures used on evaluation, we use the number of times both sets of results agree, divided by the number of results returned by the first system. Table 8 shows such a matrix of pairwise correlations. If we take the row of gold tags, the results reflect the performance of each system (the precision). In the case of MFS, the column shows that STATIC has a slightly larger correlation with the MFS than the other two methods. The matrix also shows that all our three methods agree more than 80% of the time, with PPR and STATIC having a relatively smaller agreement.

In contrast, related work using the same techniques over domain-specific words (Agirre, López de Lacalle, and Soroa 2009) shows that the results of our Personalized PageRank models departs significantly from MFS and STATIC. Table 9 shows the results of the three techniques on the three subcorpora that constitute the evaluation data set published in Koeling, McCarthy, and Carroll (2005). This data set consists of examples retrieved from the Sports and Finance sections of the Reuters corpus, and also from the balanced British National Corpus (BNC), which is used as a general domain contrast corpus.

Applying PageRank over the entire WordNet graph yields low results, very similar to those of MFS, and below those of Personalized PageRank. This confirms that STATIC PageRank is closely related to MFS, as we hypothesized in Section 5.1 and showed in Table 8 for the other general domain data sets. Whereas the results of PPR_{w2w} are very similar in the general-domain BNC, PPR_{w2w} departs from STATIC and MFS with 30 and 20 points of difference in the domain-specific Sports and Finance corpora. These results are highly relevant, because they show that PPR is able to effectively use contextual information, and depart from the MFS and STATIC baselines.

Table 8
Correlation between systems, gold tags, and MFS.

	Gold	MFS	PPR	PPR_{w2w}	STATIC
Gold	100.0	61.3	58.6	59.7	57.8
MFS	60.1	100.0	79.8	79.0	81.3
PPR	57.4	79.8	100.0	86.8	82.8
PPR_{w2w}	58.4	79.0	86.8	100.0	86.4
STATIC	56.7	81.4	82.8	86.4	100.0

Table 9
Results on three subcorpora as reported in Agirre, López de Lacalle, and Soroa (2009), where Sports and Finance are domain-specific. Best results on each column in **bold**.

System	BNC	Sports	Finance
MFS	34.9	19.6	37.1
STATIC	36.6	20.1	39.6
PPR$_{w2w}$	**37.7**	**51.5**	**59.3**

Table 10
Combination with MFS (F1). The first two rows correspond to our system with and without information from MFS. Below that we report systems that also use MFS. Best results in each column in **bold**.

System	S2AW	S3AW	S07AW	S07CG	(N)
PPR$_{w2w}$	59.7	57.9	41.7	80.1	(83.6)
PPR$_{w2w}$ MFS	**62.6**	**63.0**	48.6	81.4	(82.1)
MFS	60.1	62.3	**51.4**	78.9	(77.4)
IRST-DDD-00		58.3			
Nav05 / UOR-SSI		60.4		**83.2**	(84.1)
Ponz10				81.7	(**85.5**)

6.4.7 Combination with MFS. As mentioned in Section 6.2, we have avoided using any information regarding sense frequencies from annotated corpora, as this information is not always available for all wordnets. In this section we report the results of our algorithm when taking into account prior probabilities of senses taken from sense counts. We used the sense counts provided with WordNet in the index.sense file.[8] In this setting, the edges linking words and their respective senses are weighted according to the prior probabilities of those senses, instead of uniform weights as in Section 5.2.

Table 10 shows that results when using priors from MFS improve over the results of the original PPR$_{w2w}$ in all data sets. The improvement varies across parts of speech, and, for instance, the results for nouns in S07CG are worse (shown in rightmost column of Table 10). In addition, the results for PPR$_{w2w}$ when using MFS information improve over MFS in all cases except for S07AW.

The table also reports the best systems that do use MFS (see Section 6.3 for detailed explanations). For S2AW and S07AW we do not have references to related systems. For S3AW we can see that our system performs best. In the case of S07CG, UOR-SSI reports better results than our system. Finally, the final row reports their system when combined with MFS information as back-off (Ponzetto and Navigli 2010), which also attains better results than our system. We tried to use a combination method similar to theirs, but did not manage to improve results.

6.4.8 Efficiency of Full Graphs vs. Subgraphs. Given the very close results of our algorithm when using full graphs and subgraphs (cf. Section 6.3), we studied the efficiency of each. We benchmarked several graph-based methods on the S2AW data set, which comprises

[8] `http://wordnet.princeton.edu/wordnet/man/senseidx.5WN.html`.

2,473 instances to be disambiguated. All tests were done on a multicore computer with 16 GB of memory using a single 2.66 GHz processor. When using the full graph PPR disambiguates full sentences in one go at 684 instances per minute, whereas PPR_{w2w} disambiguates one word at a time, 70 instances per minute. The DFS subgraphs provide better performance than PPR_{w2w}, 228 instances per minute when using degree, with marginally slower performance when using PPR_{w2w} (210 instances per minute). The BFS subgraph is slowest, with around 20 instances per minute. The memory footprint of using the full graph algorithm is small, just 270 MB, so several processes can be run on a multiprocessor machine easily.

All in all, there is a tradeoff in performance and speed, with PPR_{w2w} on the full graph providing better results at the cost of some speed, and PPR on the full graph providing the best speed at the cost of worse performance. Using DFS with PPR_{w2w} lays in between and is also a good alternative, and its speed can be improved using pre-indexed paths.

6.5 Experiments on Spanish

Our WSD algorithm can be applied over non-English texts, provided that a LKB for this particular language exists. We have applied our random walk algorithms to the Spanish WordNet (Atserias, Rigau, and Villarejo 2004), using the SemEval-2007 Task 09 data set as evaluation gold standard (Màrquez et al. 2007). The data set contains examples of the 150 most frequent nouns in the CESS-ECE corpus, manually annotated with Spanish WordNet synsets. It is split into a train and test part, and has an "all words" shape (i.e., input consists of sentences, each one having at least one occurrence of a target noun). We ran the experiment over the test part (792 instances), and used the train part for calculating the MFS heuristic. The results in Table 11 are consistent with those for English, with our algorithms approaching MFS performance, and PPR_{w2w} yielding the best results. Note that for this data set the supervised algorithm could barely improve over the MFS, which performs very well, suggesting that in this particular data set the sense distributions are highly skewed.

Finally, we also show results for the first sense in the Spanish WordNet. In the Spanish WordNet the order of the senses of a word has been assigned directly by the lexicographer (Atserias, Rigau, and Villarejo 2004), as there is no information of sense frequency from hand-annotated corpora. This is in contrast to the English WordNet, where the senses are ordered according to their frequency in annotated

Table 11
Results as F1 on the Spanish SemEval07 data set, including first sense, MFS, and the best supervised system in the competition. * Statistically significant difference with respect to the best of our results (in **bold**).

Method	Acc.
PPR	78.4*
PPR_{w2w}	**79.3**
STATIC	76.5*
First sense	66.4*
MFS	84.6*
BEST	85.1*

corpora (Fellbaum 1998), and reflects the status on most other wordnets. In this case, our algorithm clearly improves over the first sense in the dictionary.

7. Conclusions

In this article we present a method for knowledge-based Word Sense Disambiguation based on random walks over relations in a LKB. Our algorithm uses the full graph of WordNet efficiently, and performs better than PageRank or degree on subgraphs (Navigli and Lapata 2007; Agirre and Soroa 2008; Navigli and Lapata 2010; Ponzetto and Navigli 2010). We also show that our combination of method and LKB built from WordNet and eXtended WordNet compares favorably to other knowledge-based systems using similar information sources (Mihalcea 2005; Sinha and Mihalcea 2007; Tsatsaronis, Vazirgiannis, and Androutsopoulos 2007; Tsatsaronis, Varlamis, and Nørvåg 2010). Our analysis shows that Personalized PageRank yields similar results when using subgraphs and the full graph, with a trade-off between speed and performance, where Personalized PageRank over the full graph is fastest, its word-to-word variant slowest, and Personalized PageRank over the subgraph lies in between.

We also show that the algorithm can be easily ported to other languages with good results, with the only requirement of having a wordnet. Our results improve over the first sense of the Spanish dictionary. This is particularly relevant for wordnets other than English. For the English WordNet the senses of a word are ordered according to the frequency of the senses in hand-annotated corpora, and thus the first sense is equivalent to the Most Frequent Sense, but this information is not always available for languages that lack large-scale hand-annotated corpora.

We have performed an extensive analysis, showing the behavior according to the parameters of PageRank, and studying the impact of different relations and WordNet versions. We have also analyzed the relation between our PPR algorithm, MFS, and STATIC PageRank. In general domain corpora they get similar results, close to the performance of the MFS learned from SemCor, but the results reported on domain-specific data sets (Agirre, López de Lacalle, and Soroa 2009) show that PPR is able to move away from the MFS and STATIC and improve over them, indicating that PPR is able to effectively use contextual information, and depart from MFS and STATIC PageRank.

The experiments in this study are readily reproducible, as the algorithm and the LKBs are publicly available.[9] The system can be applied easily to sense inventories and knowledge bases different from WordNet.

In the future we would like to explore methods to incorporate global weights of the edges in the random walk calculations (Tsatsaronis, Varlamis, and Nørvåg 2010). Given the complementarity of the WordNet++ resource (Ponzetto and Navigli 2010) and our algorithm, it would be very interesting to explore the combination of both, as well as the contribution of other WordNet related resources (Cuadros and Rigau 2008).

Acknowledgments
We are deeply indebted to the reviewers, who greatly helped to improve the article. Our thanks to Rob Koeling and Diana McCarthy for kindly providing the data set, thesauri, and assistance, and to Roberto Navigli and Simone Ponzetto for clarifying the method to map to coarse-grained senses. This work has been partially funded by the Education Ministry (project KNOW2 TIN2009-15049-C03-01).

[9] http://ixa2.si.ehu.es/ukb.

References

Agirre, E., E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies conference (NAACL/HLT'09)*, pages 19–27, Boulder, CO.

Agirre, E., T. Baldwin, and D. Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL/HLT'08)*, pages 317–325, Columbus, OH.

Agirre, E., K. Bengoetxea, K. Gojenola, and J. Nivre. 2011. Improving dependency parsing with semantic classes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT'11)*, pages 699–703, Portland, OR.

Agirre, E., O. López de Lacalle, and A. Soroa. 2009. Knowledge-based WSD on specific domains: Performing better than generic supervised WSD. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1,501–1,506, Pasadena, CA.

Agirre, E. and G. Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pages 16–22, Copenhagen.

Agirre, E. and A. Soroa. 2008. Using the multilingual central repository for graph-based word sense disambiguation. In *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC '08)*, pages 1,388–1,392, Marrakesh.

Agirre, E. and A. 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'09)*, pages 33–41, Athens.

Anaya-Sánchez, H., A. Pons-Porrata, and R. Berlanga-Llavori. 2007. TKB-UO: Using sense clustering for WSD. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 322–325, Prague.

Atserias, J., G. Rigau, and L. Villarejo. 2004. Spanish wordnet 1.6: Porting the Spanish wordnet across Princeton versions. In *Proceedings of the 4th Conference on Language Resources and Evaluation (LREC'04)*, pages 161–164, Lisbon.

Barzilay, R. and M. Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17, New York, NY.

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

Carpuat, M. and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL'07)*, pages 61–72, Prague.

Chan, Y. S. and H. T. Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*, pages 1,037–1,042, Pittsburgh, PA.

Chan, Y. S., H. T. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 33–40, Prague.

Chan, Y. S., H. T. Ng, and Z. Zhong. 2007. NUS-PT: Exploiting parallel texts for word sense disambiguation in the English all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 253–256, Prague.

Cowie, J., J. Guthrie, and L. Guthrie. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the Workshop on Speech and Natural Language (HLT'91)*, pages 238–242, Morristown, NJ.

Cuadros, M. and G. Rigau. 2006. Quality assessment of large scale knowledge resources. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, pages 534–541, Sydney.

Cuadros, M. and G. Rigau. 2007. Semeval-2007 task 16: Evaluation of wide coverage knowledge resources. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 81–86, Prague.

Cuadros, M. and G. Rigau. 2008. Knownet: Using topic signatures acquired from the Web for building automatically highly dense knowledge bases. In *Proceedings* *of the 22nd International Conference on Computational Linguistics (COLING'08)*, pages 71–84, Manchester.

Daude, J., L. Padro, and G. Rigau. 2000. Mapping WordNets using structural information. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, pages 504–511, Hong Kong.

Decadt, B., V. Hoste, W. Daelemans, and A. Van Den Bosch. 2004. GAMBL, genetic algorithm optimization of memory-based WSD. In *Proceedings of SENSEVAL-3 Third International Workshop on Evaluation of Systems for the Semantic Analysis of Text*, pages 108–112, Barcelona.

Escudero, G., L. Márquez, and G. Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC'00)*, pages 172–180, Hong Kong.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database and Some of Its Applications*. MIT Press, Cambridge, MA.

Haveliwala, T. H. 2002. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*, pages 517–526, New York, NY.

Hughes, T. and D. Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL'07)*, pages 581–589, Prague.

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

Kleinberg., J. M. 1998. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 668–677, Philadelphia, PA.

Koeling, R., D. McCarthy, and J. Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*, pages 419–426, Ann Arbor, MI.

Langville, A. N. and C. D. Meyer. 2003. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380.

Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC'86)*, pages 24–26, New York, NY.

Màrquez, L., M. A. Villarejo, T. Martí, and M. Taulé. 2007. SemEval-2007 Task 09: Multilevel semantic annotation of Catalan and Spanish. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 42–47, Prague.

McCarthy, D., R. Koeling, J. Weeds, and J. Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.

Mihalcea, R. 2002. Word sense disambiguation with pattern learning and automatic feature selection. *Natural Language Engineering*, 8:343–358.

Mihalcea, R. 2005. Unsupervised Large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT'05)*, pages 411–418, Morristown, NJ.

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

Miller, G. A., C. Leacock, R. Tengi, and R. Bunker. 1993. A semantic concordance. In *Proceedings of the Workshop on Human Language Technology (HLT'93)*, pages 303–308, Plainsboro, NJ.

Naskar, S. K. and S. Bandyopadhyay. 2007. JU-SKNSB: Extended WordNet based WSD on the English all-words task at SemEval-1. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 203–206, Prague.

Navigli, R. 2008. A structural approach to the automatic adjudication of word sense disagreements. *Natural Language Engineering*, 14(4):547–573.

Navigli, R. and M. Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 17th International Joint Conference* *on Artificial Intelligence (IJCAI'07)*, pages 1,683–1,688, Hyderabad.

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

Navigli, R., K. C. Litkowski, and O. Hargraves. 2007. SemEval-2007 Task 07: Coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 30–35, Prague.

Navigli, R. and P. Velardi. 2005. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1,075–1,086.

Navigli, Roberto, Stefano Faralli, Aitor Soroa, Oier López de Lacalle, and Eneko Agirre. 2011. Two birds with one stone: Learning semantic models for text categorization and word sense disambiguation. In *Proceedings of CIKM*, pages 2,317–2,320, Glasgow.

Ng, H. T. and H. B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, ACL '96, pages 40–47, Stroudsburg, PA.

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

Noreen, E. W. 1989. *Computer-Intensive Methods for Testing Hypotheses*. John Wiley & Sons.

Palmer, M., C. Fellbaum, S. Cotton, L. Delfs, and H. T. Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24, Toulouse.

Patwardhan, S., S. Banerjee, and T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-03)*, pages 241–257, Mexico City.

Pérez-Agüera, J. R. and H. Zaragoza. 2008. UCM-Y!R at CLEF 2008 Robust and WSD tasks. In *Proceedings of the 9th Cross Language Evaluation Forum Workshop (CLEF'08)*, pages 138–145, Aarhus.

Ponzetto, S. P. and R. Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised system. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 1,522–1,531, Uppsala.

Pradhan, S., E. Loper, D. Dligach, and M. Palmer. 2007. SemEval-2007 Task-17: English lexical sample SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 87–92, Prague.

Sinha, R. and R. Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007)*, pages 363–369, Irvine, CA.

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

Strapparava, C., A. Gliozzo, and C. Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: IRST at SENSEVAL-3. In *Proceedings of SENSEVAL-3 Third International Workshop on Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234, Barcelona.

Surdeanu, M., M. Ciaramita, and H. Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL/HLT'08)*, pages 719–727, Columbus, OH.

Sussna, M. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM'93)*, pages 67–74, New York, NY.

Tratz, S., A. Sanfilippo, M. Gregory, A. Chappell, C. Posse, and P. Whitney. 2007. PNNL: A supervised maximum entropy approach to word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 264–267, Prague.

Tsatsaronis, G., I. Varlamis, and K. Nørvåg. 2010. An experimental study on unsupervised graph-based word sense disambiguation. In *Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'10)*, pages 184–198, Iasi.

Tsatsaronis, G., M. Vazirgiannis, and I. Androutsopoulos. 2007. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1,725–1,730, Hyderabad.

Zhong, Z. and H. T. Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala.

Zhong, Z. and H. T. Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282, Jeju Island.