An Experimental Study on Unsupervised Graph-based Word Sense Disambiguation

George Tsatsaronis[1], Iraklis Varlamis[2], and Kjetil Nørvåg[1]

[1] Department of Computer and Information Science, Norwegian University of Science and Technology, {gbt,Kjetil.Norvag}@idi.ntnu.no
[2] Department of Informatics and Telematics, Harokopio University of Athens, varlamis@hua.gr

Abstract. Recent research works on unsupervised word sense disambiguation report an increase in performance, which reduces their handicap from the respective supervised approaches for the same task. Among the latest state of the art methods, those that use semantic graphs reported the best results. Such methods create a graph comprising the words to be disambiguated and their corresponding candidate senses. The graph is expanded by adding semantic edges and nodes from a thesaurus. The selection of the most appropriate sense per word occurrence is then made through the use of graph processing algorithms that offer a degree of importance among the graph vertices. In this paper we experimentally investigate the performance of such methods. We additionally evaluate a new method, which is based on a recently introduced algorithm for computing similarity between graph vertices, P-Rank. We evaluate the performance of all alternatives in two benchmark data sets, Senseval 2 and 3, using WordNet. The current study shows the differences in the performance of each method, when applied on the same semantic graph representation, and analyzes the pros and cons of each method for each part of speech separately. Furthermore, it analyzes the levels of inter-agreement in the sense selection level, giving further insight on how these methods could be employed in an unsupervised ensemble for word sense disambiguation.

1 Introduction

Word Sense Disambiguation (WSD) addresses the problem of selecting the most appropriate sense for a word, among several offered from a dictionary or a thesaurus, with respect to its context. WSD algorithms are used in several natural language processing tasks, such as machine translation, and speech processing, and the performance of the disambiguation procedure is critical to their success [6]. WSD has also been reported to boost performance of text retrieval, document classification, and document clustering tasks [13, 20]. All these findings, strengthen the need for fast and accurate WSD algorithms.

The various solutions found in the WSD bibliography face the tradeoff between unsupervised and supervised methods. The former usually offer fast execution time but low accuracy, while the latter suffer from the *knowledge acquisition bottleneck* problem because they require extensive training in a large amount of manually annotated data.

Unsupervised graph-based WSD techniques [2, 24, 35, 26, 38] have been attracting a wider focus lately, mainly because they have managed to truncate the accuracy gap from the supervised methods. The key to these methods' achievement is the rich semantic model that they employ. More specifically, they map the words to be disambiguated and their respective candidate senses to graphs, which are enhanced with nodes and semantic edges from word thesauri (e.g., WordNet). On top of this representation, they use a node ranking or node activation algorithm, which after several iterations concludes to the best candidate sense for each word, which is usually the highest ranked sense node after the convergence of the vertices' values.

In this paper, we compare the performance of several unsupervised graph-based WSD methods. We also apply for the first time a new vertices similarity measure, capitalizing on the structural similarity of the graph vertices. In the experimental evaluation we use the English WordNet [10] as our lexical database, and the data from the Senseval 2 [31] and 3 [36] *English all words* task as a benchmark. We present the comparative results of several vertex ranking algorithms [4, 8, 17], and vertex similarity algorithms [42]. The contributions of this work can be summarized in the following: (a) thorough experimental evaluation and analysis of the performance of seven state of the art unsupervised graph-based WSD methods, (b) application -for the first time- of the node similarity algorithm P-Rank [42], in the word sense disambiguation task, (c) generalized comparison and analysis against state of the art WSD approaches, both supervised and unsupervised, offering an experimental survey of the current top methods in word sense disambiguation, and (d) analysis of the methods inter-agreement in the sense selection level, that can give further insight into a possible inclusion of those methods in an ensemble of approaches.

The rest of the paper is organized as follows: Section 2 discusses the related work, and gives a short overview of the state of the art in word sense disambiguation. Section 3 presents in detail the graph construction and graph processing algorithms and their application in WSD, and also discusses the space and time complexity of the examined methods. Section 4 experimentally evaluates the compared approaches and illustrates the advantages of each method per part of speech (POS). Furthermore, it generalizes the comparison against top performing WSD methods in the Senseval 2 and 3 data sets. Finally, Section 5 concludes and provides pointers to future work.

2 Related Work

2.1 Supervised Word Sense Disambiguation

The field of WSD is a well studied research area [15, 28], mainly because the application of WSD may improve the performance of several tasks, like machine translation and text classification. A crucial component in such critical applications is the achieved accuracy of the underlying WSD system. In general, supervised WSD methods outperform their unsupervised rivals but they require extensive training in large data sets. Recent research results [28] show that the accuracy of state of the art supervised WSD methods is above 60% with an upper bound reaching 70% for all words, fine-grained WSD, while the accuracy of unsupervised methods is usually between $45 - 60\%$.

Supervised WSD approaches that report interesting performance results comprise the works of Pedersen [33], Florian et al. [11], and Carpuat et al. [40]. Pedersen uses an ensemble of 9 classifiers selected from a set of 81 Naive Bayes classifiers and requires at least one training instance for each different sense of the target word that exists in the lexicon. Similarly, Florian et al. use an ensemble of 6 different classifiers (Naive Bayes, Transformation-base learning, etc.) and report similar requirements for training samples. Carpuat et al. use a method that exploits a nonlinear kernel principal component analysis (KPCA) technique [40]. The KPCA-based model acts as the voting mechanism over a set of classifiers that learn to predict the correct sense and decides on which of the suggested senses should be selected.

State of the art results in supervised WSD have been reported by the SenseLearner system of Mihalcea and Csomai [23], the Simil-Prime system introduced by Kohomban and Lee [18], and the system developed by Hoste et al. [14]. In [23] the authors suggest the construction of seven semantic models, which are trained using the Timbl memory based learning algorithm. The Simil-Prime method [18] is trained to disambiguate words into generic semantic classes, and consequently casts the generic semantic classes back to finer grained senses, using heuristical mapping. The major drawback of this method is the use of heuristics, which cannot guarantee that finer senses will not be missed. Another drawback is the fact that it uses a decision-tree based implementation of the k-nn classifier, which raises the execution cost (mainly the space complexity) since many training examples need to be reexamined for each target word. The memory-based learning approach proposed by Hoste et al. uses voting among word-experts to decide on the correct sense. The method stores all instances in memory during training and testing, which results in both high space and time complexity.

Finally, we should mention the winners of the Senseval 2 and 3 *All English Words Task* which were the supervised WSD systems *SMUaw* [21] and *GAMBL* [9] respectively. *SMUaw* was based on pattern learning from sense-tagged corpora and instance-based learning with automatic feature selection. In the cases where the existing patterns failed to disambiguate a word and no more training data existed, the method selected the most frequent sense for the word, which resulted in high recall levels, but affected precision. In *GAMBL* word experts are trained using memory-based classifiers, that learn to predict the correct sense of each word, thus requiring extensive training.

2.2 Unsupervised Word Sense Disambiguation - The Graph-based Methods

The list of unsupervised WSD methods is long and comprises corpus-based [41], knowledge-based, such as Lesk-like [19] and graph-based [2, 24, 35, 26, 38] methods, as well as ensembles [5] that combine several methods. From all types of unsupervised WSD methods, we focus on the graph-based ones, which demonstrate high performance and seem to be a promising solution for unsupervised WSD. The first step of graph-based WSD methods relies in the construction of semantic graphs from text. The graphs are consequently processed in order to select the most appropriate meaning[3] of each examined word, in its given context.

[3] In the remaining of the paper, the words *concept*, *sense*, and *synset* may be used interchangeably to describe the meaning of a word, among the several offered by a dictionary or a word thesaurus.

One of the most influential WSD works in this direction is the disambiguation algorithm of Sussna [37], which uses the WordNet graph as a basis and examines all nouns in a window of context and assigns a sense to each noun in a way that minimizes a semantic distance function among all selected senses. In [1], Agirre and Rigau introduced and applied a similarity measure based on conceptual density between noun senses. The measure was based on WordNet's is-a hierarchy and measured the similarity between a target noun sense and the nouns in the surrounding context. More recently, Banerjee and Pedersen [3] suggested an adaptation of the original Lesk algorithm for the WordNet graph. In [24] and [38] authors use WordNet as a graph, defining the vertices as synsets and the edges as the semantic relations connecting synsets. Both methods construct synset graphs from text in the first step. Then, the former method applies the PageRank algorithm to rank the synset vertices whereas the latter employs a spreading activation technique to process the network (*SAN*) and selects the most active sense nodes after the spreading of the activation as the senses disambiguating the respective words. In [27] Navigli introduced a different graph construction method, the Structural Semantic Interconnections (SSI-HITS), in which all candidate senses are connected and consequently ranked using the HITS algorithm. SSI-HITS is based on a measure that maximizes the degree of mutual interconnection among a set of senses. Finally, in [2] Agirre and Soroa use the PageRank algorithm, instead of HITS, and a wider knowledge-base (WordNet and Extended WordNet [25] relations).

Examining unsupervised graph-based WSD from another perspective, Sinha and Mihalcea [35] propose an unsupervised graph-based method for WSD, based on an algorithm that computes graph centrality of nodes in the constructed semantic graph. To measure the centrality of the nodes, they use the indegree, the closeness, and the betweenness of the vertices in the graph, as well as PageRank. They also employ five known measures of semantic similarity or relatedness to compute the similarity of the nodes in the semantic graph, based on an idea initially presented by Patwardhan et al. [32]. Similarly, in [29], Navigli and Lapata explore several measures for analyzing the connectivity of semantic graph structures in *local* (i.e., per individual node) or *global* (i.e. for the whole graph) level. They evaluate in-degree and eigenvector centrality, maximum flow, compactness, graph entropy and edge density. They conclude that local measures perform better than global measures for the WSD task.

The examination of related literature revealed a wide variety of options in unsupervised graph-based WSD techniques. In the following of this study, we examine more closely the empirical evaluations of these methods, and analyze the reasons behind a boost in performance, which can be either the levied semantic representation or the graph processing technique itself. Furthermore, we examine the interagreement of these methods in the selection of senses when the same graph representation is employed. For this reason we implement four graph processing techniques (PageRank, SAN, HITS and P-Rank) and evaluate their performance in the same semantic representation.

3 Assigning Senses to Words in Semantic Graphs

This section presents the four graph processing methods that were selected for evaluation: SAN [8], PageRank [4], HITS [17] and P-Rank [42]. Though three of those meth-

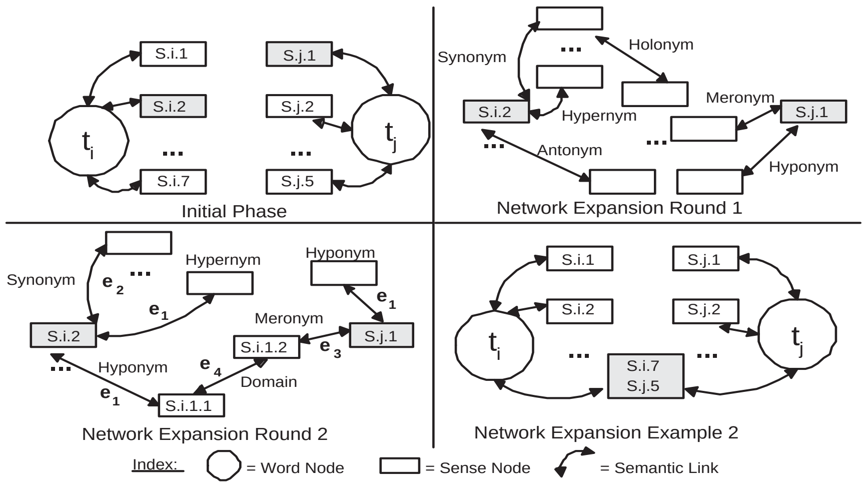

Fig. 1. Semantic Network Construction for Spreading of Activation.

ods have been applied before in WSD (SAN-based WSD [38, 39], PageRank-based WSD [2, 35], and HITS-based WSD [29]), they have never been evaluated in tandem using the same semantic representation of text. Thus, in order to provide a comparative evaluation, we used the same semantic representation (i.e., the same graph) for all methods. More specifically, we adopt the semantic network construction method that was introduced in [38]. The method utilizes all of the available semantic relations in WordNet 2.0. Furthermore, it employs a novel weighting scheme for the edges connecting the sense nodes.

3.1 Semantic Graph Construction

The semantic network construction method of Tsatsaronis et al [38] creates a semantic network for each sentence, that contains only the words that have entries in WordNet and assumes that these words have been tagged with their parts of speech (POS). The method, as depicted in Figure 1, initially adds the word nodes and their senses to the network (*initial phase*). Consequently, it adds all the thesaurus senses which are semantically related to the existing senses of the network (*expansion round 1* for the senses *S.i.2* and *S.j.1* respectively). Expansion continues iteratively until there is a path between every pair of the initial word nodes. In this step, the network ceases growing and is considered as *connected*. If there are no more senses to be expanded and the respective network is not connected, the words of that sentence cannot be disambiguated, which means a loss in coverage. Once the network is ready to be processed, the weights of each edge are added (*expansion round 2*). The weights are based on the frequency of occurrence of each edge type in the constructed network [38]. During the construction of the networks, it might be the case that two words share the same senses. This case is depicted in *expansion example 2* of Figure 1. In this case, a single sense node is added to the network (i.e., the sense nodes in the network represent WordNet synsets).

Apart from the specific method that we employ for constructing semantic graph, several other alternatives exist in the literature. In [39] the authors utilize the gloss words of the WordNet entries to construct semantic graphs. The network constructed in [24] is very similar, since it is based on some of the WordNet relations between senses, but differs in that it defines additional composite semantic relations (called *xlinks*). In [2], the authors use additional relations from the Extended WordNet and manual disambiguations of its glosses for the different entries. Finally, in [29], the network construction approach has an allowed upper bound on the length of the semantic paths. The main reason behind our selection is that the selected method incorporates all of the explicit semantic relations in WordNet and adopts an edge weighting scheme that takes into account the importance of each edge type. The selected method has been first evaluated in [38] against the approaches of [39] and [24] in graph construction and is evaluated again in this study against more recent techniques. Results show that the selected method performs better or equally well than other graph construction methods, for the same graph processing method. For example, in [38], it was compared against the representation of Veronis and Ide [39] and an accuracy improvement was reported.

3.2 Spreading of Activation (SAN) Method

The method introduced by Tsatsaronis et al. in [38], relies on spreading of activation in semantic networks (*SAN*) for WSD and it was based on an initial approach by Veronis and Ide [39] for constructing SAN for WSD. The constructed graph is processed with an iterative spreading activation strategy incorporating the fan-out and the distance constraints, as described by Crestani [8]. More specifically, the nodes initially have an activation level of 0, except for the input word nodes, whose activation is 1. In each iteration, every node propagates its activation to its neighbors, as a function of its current activation value and the weights of the edges that connect it with its neighbors. At each iteration *p* every network node *j* has an activation level $A_j(p)$ and an output $O_j(p)$, which is a function of its activation level, as shown in equation 1.

$$O_j(p) = f(A_j(p)) \tag{1}$$

The output of each node affects the next-iteration activation level of any node *k* towards which node *j* has a directed edge. Thus, the activation level of each network node *k* at iteration *p* is a function of the output, at iteration $p-1$, of every neighboring node *j* having a directed edge e_{jk}, as well as a function of the edge weight W_{jk}, as shown in equation 2. Although this process is similar to the activation spreading of feed-forward neural networks, the reader should keep in mind that the edges of SANs are bi-directional (for each edge, there exists a reciprocal edge). A further difference is that no training is involved in the case of SANs.

$$A_k(p) = \sum_j O_j(p-1) \cdot W_{jk} \tag{2}$$

Unless a function for the output *O* is chosen carefully, after a number of iterations the activation floods the network nodes. We use the function of equation 3, which incorporates fan-out and distance factors to constrain the activation spreading; τ is a threshold value.

$$O_j(p) = \begin{cases} 0 & \text{, if } A_j(p) < \tau \\ \frac{F_j}{p+1} \cdot A_j(p) & \text{, otherwise} \end{cases} \tag{3}$$

Equation 3 prohibits the nodes with low activation levels from influencing their neighboring nodes. The factor $\frac{1}{p+1}$ diminishes the influence of a node to its neighbors as the iterations progress (intuitively, as "pulses" travel further). Function F_j is a fan-out factor, defined in equation 4. It reduces the influence of nodes that connect to many neighbors.

$$F_j = (1 - \frac{C_j}{C_T}) \tag{4}$$

C_T is the total number of nodes, and C_j is the number of nodes directly connected to *j* via directed edges from j.

3.3 PageRank (PR) Method

In this work we also investigate on the potential of applying PageRank in the semantic networks shown in Figure 1. Thus, we designed another WSD algorithm (*PR*) that processes the constructed networks with PageRank. The PageRank formula that we used is a simple variation of the original PageRank equation, which takes into account edge weights as well. This variation was first introduced by Mihalcea et al. in [24]. Equation 5 shows the original PageRank formula and Equation 6 shows its weighted variation that we use to process the networks. $S(V_i)$ (and $WS(V_i)$ respectively) is the PageRank value of vertex V_i, d is the damping factor, $Out(V_j)$ is the number of outgoing links from vertex V_i and w_{ij} is the weight of the edge connecting vertices V_i and V_j.

$$S(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{S(V_i)}{|Out(V_j)|} \tag{5}$$

$$WS(V_i) = (1-d) + d \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \tag{6}$$

The *SAN* method can then be easily modified to process the constructed networks with equation 6, instead of spreading of activation. As a damping factor (d) we set 0.85, as in the original formula by Brin and Page [4], and we did not optimize this parameter. After the PageRank values stabilize, the sense nodes with the highest PagerRank scores for each target word are selected to disambiguate each word occurrence. The difference between this new PageRank-based WSD method and the method of Mihalcea et al. [24] is the semantic representation of the sentences used. In Section 4 we show that this difference in the semantic representation is important and yields an increase of almost 5% in the disambiguation accuracy. Furthermore, regarding the difference with the PageRank-based WSD algorithm introduced by Agirre and Soroa [2], this relies not only in the semantic representation of text, but also in the used PageRank formula. More specifically, Agirre and Soroa bias the PageRank execution to concentrate the initial probability mass uniformly over the word nodes that constitute the context of the word to be disambiguated.

3.4 HITS Method

In the same adopted semantic network representation we also utilize the HITS algorithm as a means of ranking the sense nodes and disambiguating text. Initially the algorithm was introduced by Kleinberg [17], and its idea is based on identifying the authorities (the most important pointed nodes in a graph) and the hubs (the nodes that point to authorities). The algorithm preceded PageRank, and it has several disadvantages, like the fact that is prone to clique-attack (i.e., densely connected neighborhoods of the graph can aggregate large scores). Its application in WSD is thus interesting, so as to investigate on how this affects the results of the task.

In HITS, each graph node has a pair of values (its hub and its authority score). Initially these values are set to 1. Then the algorithm runs in steps iteratively, to update the hub and the authority scores for each node, following the authority and the hub update rules respectively, shown in Equations 10 and 9.

$$\mathit{authority(p)} = \sum_{q \in \mathit{In(p)}} \mathit{hub(q)} \tag{7}$$

$$\mathit{hub(p)} = \sum_{r \in \mathit{Out(p)}} \mathit{authority(r)} \tag{8}$$

where *authority(i)* of a node *i* is its authority value, and *hub(i)* is its hub value, *In(i)* is the set of nodes that link to *i*, and *Out(i)* the set of nodes that *i* links to. Since our graph has edges on weights, we are using a modification of Equations 9 and 10, that take into account the edge weights. The equations are modified as follows:

$$\mathit{authority(p)} = \sum_{q \in \mathit{In(p)}} w_{q,p} \cdot \mathit{hub(q)} \tag{9}$$

$$\mathit{hub(p)} = \sum_{r \in \mathit{Out(p)}} w_{p,r} \cdot \mathit{authority(r)} \tag{10}$$

where $w_{i,j}$ is the edge weight of the edge leaving from i and linking to j. Eventually, after a large number of iterations, the authority and the hub values may converge if a normalization is used, which divides at each step each authority value by the sum of the authority values and each hub value by the sun of the hub values. In practice, we are using a small threshold (i.e., 10^{-4}) which acts as a criterion of change from step to step during the iterations, and when the changes affecting the authority and the hub values do not surpass it for any node in the graph, we assume that the values have converged. Eventually, the sense node with the highest authority value is selected as the most appropriate sense for each word.

3.5 P-Rank Method

The P-Rank measure [42] (Penetrating Rank) is a very recently introduced measure of structural similarity for nodes in an information network. It enriches a former successful measure of node similarity in information networks, SimRank [16]. In their paper, the authors prove that P-Rank is a unified structural similarity network, under which all state of the art similarity measures, including CoCitation, Coupling, Amsler and SimRank, are just its special cases. In this work, it is for the first time that P-Rank is applied for WSD. The basic idea behind P-Rank is that two vertices in an information network are similar, if they are referenced by similar vertices, and they also reference similar vertices. P-Rank is recursive, and it executes over all pairs of vertices in a given graph. Let a graph G and two vertices a, b. Also let $R_k(a,b) = R_k(b,a)$ denote the P-Rank similarity value for the pair of vertices (a,b), at iteration k. Then, P-Rank can be formalized as shown in Equation 11:

$$\begin{aligned} R_{k+1}(a,b) = \lambda \cdot \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b)) \\ + (1-\lambda) \cdot \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} R_k(O_i(a), O_j(b)) \end{aligned} \tag{11}$$

where $I(n)$ of a vertex n is the set of its incoming neighbors, $I_i(n)$ is the i_{th} element of this set, and the respective holds for the $O(n)$ notation. The parameter $\lambda \in [0,1]$ balances the relative weight of in- and out-link directions, and is usually set to 0.5. $C \in [0,1]$ is a damping factor for in- and out-link directions, usually taking the value of 0.8, according to the authors. Finally, the number of iterations needed, for the vertex pairs similarity values to converge, is reported to be empirically at $ln(n)$, where n is the number of vertices in the graph. Since our semantic networks have weights on their edges, we are using a modification of Equation 11 to accommodate our weighting scheme. Thus, we modify the definition of $|I(a)|$, and $|O(a)|$ of a vertex a, as follows:

$$|I(a)| = \sum_{i \in \textit{Incoming(a)}} w_{i,a} \tag{12}$$

$$|O(a)| = \sum_{j \in \textit{Outgoing(a)}} w_{a,j} \tag{13}$$

where *Incoming(a)* and *Outgoing(a)* are the lists of the incoming and outgoing neighbors of *a*. Then, the sums in equation 11 are of course modified to run over the respective $|$*Incoming(a)*$|$ and $|$*Outgoing(a)*$|$. After the convergence of the similarity values between all pairs of vertices, the correct sense for each word is the sense node having the highest similarity with the respective word node in our networks.

3.6 WSD Methods Complexity

With regards to the complexity of the four methods, in [38] it was shown that the construction time of the semantic networks is $O(n \cdot k^{l+1})$ where n is the number of words we disambiguate, k is the maximum branching factor of the used thesaurus nodes and l is the maximum semantic path length in the thesaurus. The time complexity of *SAN* is $O(n^2 \cdot k^{2l+3})$. The time complexity of *PR* is $O(n^2 \cdot k^{\frac{3}{2}l+3})$ in the worst case, where the network has $n \cdot k^{\frac{l}{2}+1}$ nodes and $n \cdot k^{l+2}$ edges, and similar is the time complexity of the

	Senseval 2					***Senseval 3***				
	N	**V**	**Adj.**	**Adv.**	**All**	**N**	**V**	**Adj.**	**Adv.**	**All**
Mono.	260	33	80	91	464	193	39	72	13	317
Poly.	813	502	352	172	1839	699	686	276	1	1662
Av. Poly.	4.21	9.9	3.94	3.23	5.37	5.07	11.49	4.13	1.07	7.23
Av. Poly. (P. only)	5.24	10.48	4.61	4.41	6.48	6.19	12.08	4.95	2.0	8.41

Table 1. Occurrences of polysemous and monosemous noun (N), verb (V), adjective (Adj.), adverb (Adv.) and total (All) words of WordNet 2 in Senseval 2, and 3.

HITS and the method. The time complexity of P-Rank in the worst case is even larger; $O(n^4)$ [42], since it runs over all pair combinations of vertices. Its space complexity though, is the same with the rest of the algorithms. The space complexity at the worst case is equal to the complexity required in memory to construct the semantic networks, and for the disambiguation of n words is equal to $O(n^2 \cdot k^{2l+3})$.

4 Experimental Evaluation

In this section we proceed with an empirical evaluation of the performance of the four methods, which examines two criteria: (1) the accuracy of the methods in two benchmark data sets, and (2) the inter-agreement rate of the methods in the sense selection level, in the same data sets. In order to evaluate the examined methods we use the Senseval 2 [31] and 3 [36] *All English Words Task* data sets for testing. These data sets were manually annotated with the correct senses by human annotators, before the respective competitions were conducted.[4] In Table 1 we present the statistics of those data sets, including average polysemy of words, both with (Av. Poly.) and without (Av. Poly. (P. only)) taking into account monosemous words. Senseval 2 is easier to disambiguate than Senseval 3, as the average polysemy is larger in the latter. Adverbs are very easy to disambiguate and are usually excluded from the evaluation (e.g., Senseval 3 has only 13 adverb occurrences with average polysemy close to 1). The verb POS is the most difficult to disambiguate, since a typical verb has more than 8 different senses from WordNet.

Regarding the lower and upper bounds of WSD methods in those data sets, a straightforward lower bound is to select randomly a sense for each word occurrence. This disambiguation method would produce an accuracy of around 20% for Senseval 2 and SemCor, and 14% for Senseval 3. A reasonable upper bound, as stated in [28], would be the interannotator agreement or intertagger agreement (ITA), that is, the percentage of words tagged with the same sense by two or more human annotators. The interannotator agreement on coarse-grained (lexicons with few and clearly distinct senses for each lemma are used), possibly binary (two senses per lemma), sense inventories is calculated around 90% [12, 29], whereas on fine-grained, WordNet-style sense inventories, where there are many senses per lemma and which are often hard to distinguish, the inter-annotator agreement is estimated between 67% and 80% [7, 30, 36].

[4] http://www.senseval.org/

Method	Senseval 2				Senseval 3			
	N	V	Adj.	All	N	V	Adj.	All
SAN	53.9	31.7	59.0	49.5	50.8	36.5	58.0	46.8
PR	69.5	37.2	59.0	58.8	61.8	47.3	60.6	56.7
HITS	69.1	36.6	59.1	58.3	69.2	40.4	66.7	57.4
P-Rank	51.3	27.31	57.4	45.6	60.6	29.9	67.8	52.1
Mih05	57.5	36.5	56.7	52.0	*n/a*	*n/a*	*n/a*	51.8
Agi09	70.4	38.9	58.3	59.5	64.1	46.9	62.6	57.4
Nav07	*n/a*	*n/a*	*n/a*	*n/a*	61.9	36.1	62.8	52.5
FS	74.0	42.4	63.1	63.7	70.9	50.7	59.7	61.3

Table 2. Overall and per POS accuracies (%) of WSD methods in Senseval 2, and 3 (*All English Words Task* data sets) for all POS, excluding adverbs.

4.1 Empirical Evaluation of Unsupervised Graph-based WSD methods

Table 2 shows the accuracy of the four methods for all POS in the two data sets. We have also added in the comparison, results from related methods with regards to unsupervised graph-based WSD. These are: the method of Mihalcea et al. [22] (Mih05), the method of Agirre and Soroa [2] (Agi09), and the results from the work of Navigli and Lapata [29] (Nav07). For this latter work, because the authors test and compare several graph connectivity measures, the table contains the numbers of their *KPP* measure, which was shown by their analysis to be the best performing graph-based measure overall. Note also, that adverbs are omitted in the comparison, since they are very few in number in the Senseval competitions, compared to the rest POS. Whenever results were not available, because they were not reported in the literature, an entry *n/a* exists in the respective cell. Finally, we have also added in the comparison a simple heuristic method (*FS*) that always selects the first sense of the target word from WordNet (i.e., the most frequent) to conduct the disambiguation. Though this method is usually reported as a baseline for the supervised systems (the unsupervised systems' baseline is the random assignment of senses), we have added it into the comparison, so that practitioners of WSD have a clear idea of the performance the unsupervised WSD systems can offer against the supervised ones.

As Table 2 shows the *SAN* method has stable performance, obtaining an accuracy very near 50%, overall for all POS. The *PR* method shows impressive increase in accuracy over the method of Mihalcea et al., which is due to the different semantic representation used through the constructed semantic networks, since the PageRank formula remaind the same in both cases. The *HITS* method performs overall better than *SAN* and its performance is very close to the *PR* method. In fact, *HITS* seems to be performing equally (Senseval 2) or better (Senseval 3) for the noun POS than *PR*, and the same holds for the adjective POS. For the verb POS, *PR* performs overall better than *HITS*. The *P-Rank* method does not seem to perform very well against the rest unsupervised graph-based techniques, but this is in accordance with the results reported by Navigli and Lapata [29], who reported lower results than other graph-based methods for the *betweenness* and the *indgree* measures of structural similarity in semantic graphs. The method of Agirre and Soroa also performs very well, and is in fact the best unsupervised graph-based method in Senseval 2, and has the same performance with *HITS*

Pair	Senseval 2				Senseval 3			
	N	**V**	**Adj.**	**All**	**N**	**V**	**Adj.**	**All**
SAN - PR	51.51	35.74	54.16	47.86	53.17	49.48	49.83	51.21
SAN - HITS	52.42	23.89	57.55	39.51	50.6	40.38	50.16	46.68
SAN - P-Rank	50.84	27.16	63.46	46.77	66.52	32.94	69.04	55.37
PR - HITS	62.56	34.93	64.32	55.54	60.36	44.64	66.88	55.57
PR - P-Rank	50.55	30.95	67.3	48.1	68.2	30.58	71.42	55.78
HITS - P-Rank	53.88	23.8	59.61	46.83	67.78	31.76	69.04	54.17

Table 3. *SAN*, *PR*, *HITS*, and *P-Rank* methods' pairwise inter-agreement ($\%$) in Senseval 2 and 3 (*All English Words Task* data sets) for all POS, excluding adverbs.

in the Senseval 3 data set. The performance difference between these two methods is not statistically significant at the 0.95 confidence level, if one examines their overall accuracy in the respective data sets. The *KPP* measure of Navigli and Lapata cannot match the accuracy of *PR*, *HITS* and the method of Agirre and Soroa. Regarding the *FS* method, though simple, outperforms every other compared method. It obtains very high accuracies, always above 60% for all POS. Its performance in nouns and adjectives is impressive, but in the verbs, due to their large average polysemy, the performance drops dramatically, compared to the rest POS. In another interesting unsupervised approach, Pedersen and Kolhatkar [34] perform disambiguation in Senseval 2 and 3, using measures of semantic relatedness. Their best reported results in F-Measure were 59% for Senseval 2 and 54% for Senseval 3, performance which is almost the same with PR and HITS in Senseval 2, but slightly worse in Senseval-3.

One additional comparison we would like to make regarding the four studied methods (*SAN*, *PR*, *HITS*, and *P-Rank*) is to examine the percentage of times the four methods agree in the sense selection level. Previous studies have shown that the ensemble of methods can lead to increased WSD accuracy [5]. A prerequisite is that the methods do not agree very often, so that there is a potential benefit from the ensemble. In this direction, we have measured their pair-wise inter-agreement rate (i.e., the percentage of the same sense assignment to the total sense assignments performed). Table 3 shows the inter-agreement rate for all pairwise combinations of the four methods, separately for each POS, and for each of the two examined data sets. The aim of this analysis is to investigate whether a potential combination of any subset of the four methods in an ensemble of unsupervised methods (e.g., [5]), would be expected to yield interesting results. The performance of the ensemble is strongly related to the pluralism of suggestions of the underlying WSD methods. As the table shows, the pairwise inter-agreement rate of the four methods is always lower or very close to 70% in all cases. The lowest inter-agreement rates are reported for the verb POS, which is an expected outcome, since the verbs are more polysemous than the rest POS. The lowest inter-agreement rates are reported for the *SAN-PR* and *SAN-HITS* pairs. This means that in a possible ensemble, the combination of *SAN* with *PR* or *HITS* could boost the overall performance. In parallel, we can observe from the table that all the methods seem to agree more in a pairwise manner in Senseval 3 than in Senseval 2. This is an interesting finding, because Senseval 3 is more polysemous than Senseval 2, and maybe the reverse was expected. A possible interpretation is that in the case of Senseval 3 the networks are larger, and thus more densely connected, and so the applied measures recognize more

Dataset	SenseLearner	Simil-Prime	SSI	WE	FS	PR	HITS	Agi09
Senseval2	64.82	65.00	n/a	63.2	63.7	58.8	58.3	59.5
Senseval3	63.01	65.85	60.4	n/a	61.3	56.7	57.4	57.4

Table 4. Accuracies (%) on Senseval 2 and 3 *All English Words Task* data sets, excluding adverbs.

easily the most important vertices. Overall, these findings show that the combination of these four graph-based measures has great potential due to their relatively low level of inter-agreement.

4.2 Comparison with State of the Art WSD Methods

In this section we generalize the comparison of the unsupervised graph-based WSD methods with the state of the art results reported in the WSD literature, independently of the type of methods used. In this direction, we compare the top 3 methods from Table 2, namely *PR*, *HITS* and the method of Agirre and Soroa, with the highest results reported i the WSD literature for Senseval 2 and 3. Thus, we compare with the methods of Mihalcea and Csomai [23] (SenseLearner), Kohomban and Lee [18] (Simil-Prime), Navigli [26] (SSI), and Hoste et al. [14] (WE), in the Senseval 2 and 3 data sets. Table 4.2 shows the respective accuracies, where available. We can also refer to the unsupervised enseble method of Brody et al. [5] only in the noun POS of Senseval 3 data set, since their method's evaluation is limited to that. Brody et al. report an accuracy of 63.9% in Senseval 3 nouns (Senseval 2 is N/A) with an upper bound of their ensemble close to 70%. From the results of Table 4.2 we can observe that the top performing method appears to be the *Simil-Prime*, with very high overall accuracy, equal or above to 65%. We have to note though, that the latter cannot disambiguate adjectives and adverbs, residing in the FS method to perform the task for these two POS. It is also obvious from the results table, that though the unsupervised graph-based WSD techniques cannot match the accuracy of the rest of the methods, they have clearly closed the gap very much towards a possible match in the near future.

5 Conclusions and Future Work

In this work we presented an experimental study of unsupervised graph-based WSD techniques. The aim was to analyze the performance of known techniques for processing semantic graphs, keeping the same semantic representation, so that the comparison is compatible. In our comparison we included a spreading of activation method, the PageRank and HITS algorithms, as well as, for the first time, the P-Rank structural similarity measure for vertices in information networks. A thorough experimental evaluation was conducted in two benchmark WSD data sets. We also compared against other known unsupervised graph-based WSD techniques, that do not use the same semantic representation, as well as against the top reported results in the two data sets. Furthermore, we analyzed the pairwise inter-agreement rate between the examined methods, and we showed that it is low for most pairs, leading to the conclusion that several of these methods can form up an ensemble. Our study also showed that the gap in accuracy between supervised methods and unsupervised graph-based techniques, has been truncated over the last years, constituting a solid evidence that there is still room for improvement, given also the fact that thesauri, like WordNet, will keep developing. In the future we plan to design unsupervised ensembles of graph-based methods, taking advantage of the relatively low inter-agreement rate and aiming at a high accuracy learner for the task.

References

1. E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proc. of COLING*, pages 16–22, 1996.
2. E. Agirre and A. Soroa. Personalizing pagerank for word sense disambiguation. In *Proc. of EACL*, pages 33–41, 2009.
3. S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proc. of IJCAI*, 2003.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:1–7, 1998.
5. S. Brody, R. Navigli, and M. Lapata. Ensemble methods for unsupervised wsd. In *Proc. of COLING/ACL 2006*, pages 97–104, 2006.
6. Y. Chan, H. Ng, and D. Chiang. Word sense disambiguation improves statistical machine translation. In *Proc. of ACL*, 2007.
7. T. Chklovski and R. Mihalcea. Exploiting agreement and disagreement of human annotators for word sense disambiguation. In *Proc. of RANLP*, 2003.
8. F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11:453–482, 1997.
9. B. Decadt, V. Hoste, W. Daelemans, and A. van den Bosch. Gambl, genetic algorithm optimization of memory-based wsd. In *Proc. of the Senseval3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.
10. C. Fellbaum. *WordNet – an electronic lexical database*. MIT Press, 1998.
11. R. Florian, S. Cucerzan, C. Schafer, and D. Yarowsky. Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4):327–341, 2002.
12. W. Gale, K. Church, and D. Yarowsky. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proc. of the ACL 1992*, pages 249–256, 1992.
13. J. Gonzalo, F. Verdejo, and I. Chugur. Indexing with wordnet synsets can improve text retrieval. In *Proc. of the COLING/ACL Workshop on Usage of WordN et for NLP*, 1998.
14. V. Hoste, W. Daelemans, I. Hendrickx, and A. van den Bosch. Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation. In *Proc. of the ACL Workshop on Word Sense Disambiguation*, 2002.
15. N. Ide and J. Veronis. Word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40, 1998.
16. G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proc. of KDD*, pages 538–543, 2002.
17. J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
18. U. Kohomban and W. Lee. Learning semantic classes for word sense disambiguation. In *Proc. of ACL*, pages 34–41, 2005.
19. M. Lesk. Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proc. of the SIGDOC Conference*, pages 24–26, 1986.

20. D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word sense disambiguation for exploiting hierarchical thesauri in text classification. In *Proc. of PKDD*, pages 181–192, 2005.
21. R. Mihalcea. Word sense disambiguation with pattern learning and automatic feature selection. *Natural Language Engineering*, 1(1):1–15, 2002.
22. R. Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT*, 2005.
23. R. Mihalcea and A. Csomai. Senselearner: Word sense disambiguation for all words in unrestricted text. In *Proc. of ACL*, pages 53–56, 2005.
24. R. Mihalcea, P. Tarau, and E. Figa. Pagerank on semantic networks with application to word sense disambiguation. In *Proc. of COLING*, 2004.
25. D. Moldovan and V. Rus. Logic form transformation of wordnet and its applicability to question answering. In *Proc. of ACL*, pages 394–401, 2001.
26. R. Navigli. Online word sense disambiguation with structural semantic interconnections. In *Proc. of EACL*, 2006.
27. R. Navigli. A structural approach to the automatic adjudication of word sense disagreements. *Natural Language Engineering*, 14(4):547–573, 2008.
28. R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2), Article 10, 2009.
29. R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In *Proc. of IJCAI*, pages 1683–1688, 2007.
30. M. Palmer, H. Dang, and C. Fellbaum. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Journal of Natural Language Engineering*, 13(2):137–163, 2007.
31. M. Palmer, C. Fellbaum, and S. Cotton. English tasks: All-words and verb lexical sample. In *Proc. of Senseval-2*, pages 21–24, 2001.
32. S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proc. of CICLing*, pages 241–257, 2003.
33. T. Pedersen. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *Proc. of NAACL*, pages 63–69, 2000.
34. T. Pedersen and V. Kolhatkar. WordNet::SenseRelate::AllWords - A Broad Coverage Word Sense Tagger that Maximimizes Semantic Relatedness. In *Proc. of NAACL/HLT*, pages 17–20, 2009.
35. R. Sinha and R. Mihalcea. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proc. of ICSC*, 2007.
36. B. Snyder and M. Palmer. The english all-words task. In *Proc. of Senseval-3*, pages 41–43, 2004.
37. M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proc. of CIKM*, 1993.
38. G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proc. of IJCAI*, pages 1725–1730, 2007.
39. J. Veronis and N. Ide. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proc. of COLING*, pages 389–394, 1990.
40. D. Wu, W. Su, and M. Carpuat. A kernel pca method for superior word sense disambiguation. In *Proc. of ACL*, pages 637–644, 2004.
41. D. Yarowsky. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of COLING*, pages 454–460, 1992.
42. P. Zhao, J. Han, and Y. Sun. P-Rank: a comprehensive structural similarity measure over information networks. In *Proc. of CIKM*, pages 553–562, 2009.