Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, 2007, pp. 1725-1730.

Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri

George Tsatsaronis[1*], **Michalis Vazirgiannis**[1,2†] and **Ion Androutsopoulos**[1]
[1]Department of Informatics, Athens University of Economics and Business, Greece
[2]GEMO Team, INRIA/FUTURS, France

Abstract

Most word sense disambiguation (WSD) methods require large quantities of manually annotated training data and/or do not exploit fully the semantic relations of thesauri. We propose a new unsupervised WSD algorithm, which is based on generating Spreading Activation Networks (SANs) from the senses of a thesaurus and the relations between them. A new method of assigning weights to the networks' links is also proposed. Experiments show that the algorithm outperforms previous unsupervised approaches to WSD.

1 Introduction

Word Sense Disambiguation (WSD) aims to assign to every word of a document the most appropriate meaning (sense) among those offered by a lexicon or a thesaurus. WSD is important in natural language processing and text mining tasks, such as machine translation, speech processing, information retrieval, and document classification. A wide range of WSD algorithms and techniques has been developed, utilizing machine readable dictionaries, statistical and machine learning methods, even parallel corpora. In [Ide and Veronis, 1998] several approaches are surveyed; they address WSD either in a supervised manner, utilizing existing manually-tagged corpora, or with unsupervised methods, which sidestep the tedious stage of constructing manually-tagged corpora.

The expansion of existing, and the development of new word thesauri has offered powerful knowledge to many text processing tasks. For example, exploiting is-a relations (e.g. hypernyms/hyponyms) between word senses leads to improved performance both in text classification [Mavroeidis *et al.*, 2005] and retrieval [Voorhees, 1993]. Semantic links between senses, as derived from a thesaurus, are thus important, and they have been utilized in many previous WSD approaches, to be discussed below. Previous WSD approaches, however, have not considered the full range of semantic links between senses in a thesaurus. In [Banerjee and Pedersen, 2003] a larger subset of semantic relations compared to previous approaches was used, but antonymy, domain/domain terms and all inter-POS relations were not considered.

In this paper we propose a new WSD algorithm, which does not require any training, in order to explore the aforementioned potential. A word thesaurus is used to construct Spreading Activation Networks (SANs), initially introduced in WSD by Quillian [1969]. In our experiments we used WordNet [Fellbaum, 1998], though other thesauri can also be employed. The innovative points of this new WSD algorithm are: (a) it explores all types of semantic links, as provided by the thesaurus, even links that cross parts of speech, unlike previous knowledge-based approaches, which made use of mainly the "is-a" and "has-part" relations; (b) it introduces a new method for constructing SANs for the WSD task; and (c) it introduces an innovative weighting scheme for the networks' edges, taking into account the importance of each edge type with respect to the whole network. We show experimentally that our method achieves the best reported accuracy taking into account all parts of speech on a standard benchmark WSD data set, Senseval 2 [Palmer *et al.*, 2001].

The rest of the paper is organized as follows. Section 2 provides preliminary information concerning the use of semantic relations and SANs in the WSD task. Section 3 discusses our SAN construction method and edge weighting scheme. Section 4 presents our full WSD method, and Section 5 experimental results and analysis. Section 6 discusses related WSD approaches, and Section 7 concludes.

2 Background

2.1 Semantic Relations in WordNet

WordNet is a lexical database containing English nouns, verbs, adjectives and adverbs, organized in synonym sets (synsets). Hereafter, the terms *senses* and *synsets* are used interchangeably. Synsets are connected with various edges, representing semantic relations among them, and the latest WordNet versions offer a rich set of such links: hypernymy/hyponymy, meronymy/holonymy, synonymy/antonymy, entailment/causality, troponymy, domain/domain terms, derivationally related forms, coordinate terms, attributes, and stem adjectives. Several relations cross parts of speech, like the *domain terms* relation, which connects senses pertaining to the same domain (e.g. *light*, as a

*Partially funded by the PENED 2003 Programme of the EU and the Greek General Secretariat for Research and Technology.
†Supported by the Marie Curie Intra-European Fellowship.

noun meaning electromagnetic radiation producing a visual sensation, belongs to the domain of *physics*), and the *attributes* relation, which connects a word with its possible values (e.g. *light*, as a noun, can be *bright* or *dull*). Our method utilizes all of the aforementioned semantic relations.

2.2 WSD Methods that Exploit Semantic Relations

Several WSD approaches capitalize on the fact that thesauri like WordNet offer important vertical ("is-a", "has-part") and horizontal (synonym, antonym, coordinate terms) semantic relations. Some examples of these approaches are [Sussna, 1993; Rigau *et al.*, 1997; Leacock *et al.*, 1998; Mihalcea *et al.*, 2004; Mavroeidis *et al.*, 2005; Montoyo *et al.*, 2005]. Most of the existing WSD approaches, however, exploit only a few of the semantic relations that thesauri provide, mostly synonyms and/or hypernyms/hyponyms. For example Patwardhan et al. [2003] and Banerjee and Pedersen [2003] focus mostly on the "is-a" hierarchy of nouns, thus ignoring other intra- and all inter-POS relations. In contrast, our method takes into account all semantic relations and requires no training data other than the thesaurus.

2.3 Previous Use of SANs in WSD

Spreading Activation Networks (SANs) have already been used in information retrieval [Crestani, 1997] and in text structure analysis [Kozima and Furugori, 1993]. Since their first use in WSD by Quillian [1969], several others [Cotrell and Small, 1983; Bookman, 1987] have also used them in WSD, but those approaches required rather ad hoc hand-encoded sets of semantic features to compute semantic similarities. The most recent attempt to use SANs in WSD that we are aware of, overcoming the aforementioned drawback, is [Veronis and Ide, 1990].

Figure 1 illustrates how SANs were applied to WSD by Veronis and Ide. Let W_1 and W_2 be two words that co-occur (e.g. in a sentence or text) and that we want to disambiguate. They constitute the network nodes (word nodes) depicted in the *initial phase* of Figure 1; more generally, there would be n word nodes, corresponding to the n words of the text fragment. Next, all relevant senses of W_1 and W_2 are retrieved from a machine readable dictionary (MRD), and are added as nodes (sense nodes) to the network. Each word is connected to all of its senses via edges with positive weights (activatory edges). The senses of the same word (e.g. S_{11} and S_{12}) are connected to each other with edges of negative weight (inhibitory edges). This is depicted as *phase 1* in Figure 1. Next, the senses' glosses are retrieved, tokenized, and reduced to their lemmas (base forms of words). Stop-words are removed. Each gloss word (GW) is added as a node to the network, and is connected via activatory links to all sense nodes that contain it in their glosses (*phase 2*). The possible senses of the gloss words are retrieved from the MRD and added to the network. The network continues to grow in the same manner, until nodes that correspond to a large part or the whole of the thesaurus have been added. Note that each edge is bi-directional, and each direction can have a different weight.

Once the network is constructed, the initial word nodes are activated, and their activation is spread through the network according to a spreading strategy, ensuring that eventually only one sense node per initial word node will have a positive activation value, which is taken to be the sense the algorithm assigns to the corresponding word. Note that this approach assumes that all occurrences of the same word in the text fragment we apply the algorithm to have the same sense, which is usually reasonable, at least for short fragments like sentences or paragraphs.

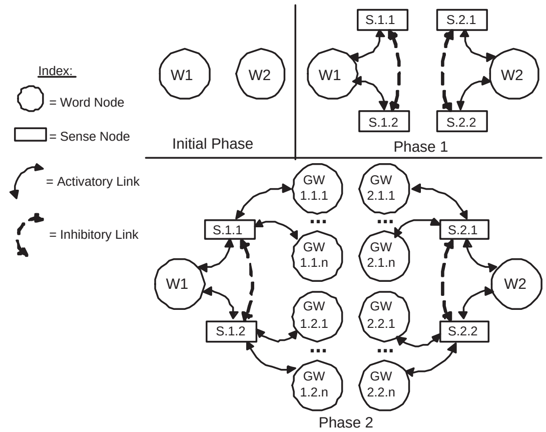

Figure 1: A previous method to generate SANs for WSD.

We use a different method to construct the network, explained below. Note also that Veronis and Ide do not use direct sense-to-sense relations; in contrast, we use all such relations, as in section 2.1, that exist in the thesaurus. Moreover they weigh all the activatory and inhibitory edges with 1 and -1, whereas we propose a new weighting scheme, taking into account the importance of each edge type in the network.

3 SAN Creation

In this section we propose a new method to construct SANs for the WSD task, along with a new weighting scheme for the edges. Our algorithm disambiguates a given text sentence by sentence. We only consider the words of each sentence that are present in the thesaurus, in our case WordNet. We also assume that the words of the text have been tagged with their parts of speech (POS). For each sentence, a SAN is constructed as shown in Figure 2. For simplicity, in this example we kept only the nouns of the input sentence, though the method disambiguates all parts of speech. The sentence is from the *d00* file in Senseval 2 data set:

> "If both **copies** of a certain **gene** were knocked out, benign **polyps** would develop."

To construct the SAN, initially the word nodes, in our case the nouns *copies*, *gene* and *polyps*, along with their senses are added to the network, as shown in the *initial phase* of Figure 2. The activatory and inhibitory links are then added, as in the previous section, but after this point the SAN grows in a very different manner compared to Veronis and Ide. First, all the senses of the thesaurus that are directly linked to the existing senses of the SAN via any semantic relation are added to the SAN, along with the corresponding links, as shown in *expansion round 1* of Figure 2. Every edge is bi-directional, since

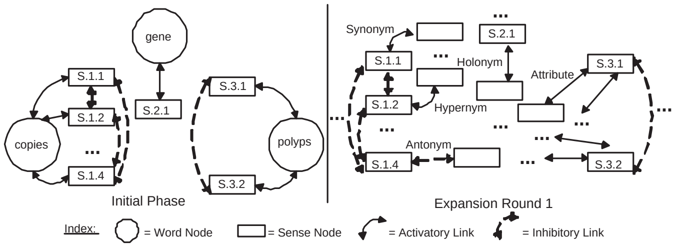

Figure 2: Our method to construct SANs.

the semantic relations, at least in WordNet, are bi-directional (e.g. if S_1 is a hypernym of S_2, S_2 is a hyponym of S_1). In the next expansion round, the same process continues for the newly added sense nodes of the previous round. The network ceases growing when there is a path between every pair of the initial word nodes. Then the network is considered as *connected*. If there are no more senses to be expanded and the respective SAN is not connected, we cannot disambiguate the words of that sentence, losing in coverage. Note that when adding synsets, we use breadth-first search with a closed set, which guarantees we do not get trapped into cycles.

3.1 The Spreading Activation Strategy

The procedure above leads to networks with tens of thousands of nodes, and almost twice as many edges. Since each word is eventually assigned its most active sense, great care must be taken in such large networks, so that the activation is efficiently constrained, instead of spreading all over the network.

Our spreading activation strategy consists of iterations. The nodes initially have an activation level 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. We adopt the activation strategy introduced by Berger et al. [2004], modifying it by inserting a new scheme to weigh the edges, which is discussed in section 3.2. More specifically, 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.

3.2 Assigning Weights to Edges

In information retrieval, a common way to measure a token's importance in a document is to multiply its term frequency in the document (*TF*) with the inverse (or log-inverse) of its document frequency (*IDF*), i.e. with the number of documents the token occurs in. To apply the same principle to the weighting of SAN edges, we consider each node of a SAN as corresponding to a document, and each type of edge (each kind of semantic relation) as corresponding to a token.

Initially each edge of the SAN is assigned a weight of -1 if it is inhibitory (edges representing antonymy and competing senses of the same word), or 1 if it is activatory (all other edges). Once the network is constructed, we multiply the initial weight w_{kj} of every edge e_{kj} with the following quantity:

$$ETF(e_{kj}) \cdot INF(e_{kj}) \tag{5}$$

ETF, defined in equation 6, is the edge type frequency, the equivalent of *TF*. It represents the percentage of the outgoing edges of *k* that are of the same type as e_{kj}. When computing the edge weights, edges corresponding to hypernym and hyponym links are considered of the same type, since they are symmetric. The intuition behind *ETF* is to promote edges whose type is frequent among the outgoing edges of node *k*, because nodes with many edges of the same type are more likely to be hubs for the semantic relation that corresponds to that type.

$$ETF(e_{kj}) = \frac{|\{e_{ki}|type(e_{ki}) = type(e_{kj})\}|}{|\{e_{ki}\}|} \tag{6}$$

The second factor in equation 5, defined in equation 7, is the inverse node frequency (*INF*), inspired by *IDF*. It is the frequency of e_{kj}'s type in the entire SAN.

$$INF(e_{kj}) = \log \frac{N+1}{N_{type(e_{kj})}} \tag{7}$$

N is the total number of nodes in the SAN, and $N_{type(e_{kj})}$ is the number of nodes that have outgoing edges of the same type as e_{kj}. As in *IDF*, the intuition behind *INF* is that we want to promote edges of types that are rare in the SAN.

4 The WSD Algorithm

Our WSD algorithm consists of four steps. Given a POS-tagged text, a designated set of parts of speech to be disambiguated, and a word thesaurus:

Step 1: Fragment the text into sentences, and select the words having a part of speech from the designated set. For each sentence repeat steps 2 to 4.

Step 2: Build a SAN, according to section 3. If the SAN is not connected, even after expanding all available synsets, abort the disambiguation of the sentence.

Step 3: Spread the activation iteratively until all nodes are inactive.[1] For every word node, store the last active sense node with the highest activation.[2]

Step 4: Assign to each word the sense corresponding to the sense node stored in the previous step.

5 Experimental Evaluation

We evaluated our algorithm on a major benchmark WSD data set, namely Senseval 2 in the "*English all words*" task. The data set is annotated with senses of WordNet 2. We experimented with all parts of speech, to be compatible with all published results of Senseval 2 [Palmer *et al.*, 2001]. Table 1 shows the number of occurrences of polysemous and monosemous words of WordNet 2 in the data set we used, per POS, as well as the average polysemy.

	Nouns	**Verbs**	**Adj.**	**Adv.**	**Total**
Monosemous	260	33	80	91	464
Polysemous	813	502	352	172	1839
Av. Polysemy	4	9	3	3	5

Table 1: Occurrences of polysemous and monosemous words of WordNet 2 in Senseval 2.

5.1 Methods Compared

In order to compare our WSD method to that of [Veronis and Ide, 1990], we implemented the latter and evaluated it on Senseval 2, again using WordNet 2. We also included in the comparison the baseline for unsupervised WSD methods, i.e. the assignment of a random sense to each word. For the baseline, the mean average of 10 executions is reported. Moreover, in order to evaluate the possibility of including glosses in our method, instead of only synset-to-synset relations, we implemented a hybrid method which utilizes both, by adding to our SANs the gloss words of the synsets along with their senses, similarly to the method of Veronis and Ide. For the purposes of this implementation, as well as for the implementation of the original method of Veronis and Ide, we used the Extended WordNet [Moldovan and Rus, 2001], which provides the POS tags and lemmas of all WordNet 2 synset glosses. In the comparison, we also include the results presented in [Mihalcea *et al.*, 2004]. There, another unsupervised knowledge-based method is proposed and is evaluated on Senseval 2; it uses thesauri-generated semantic networks, along with Pagerank for their processing. We also report the accuracy of the best reported unsupervised method that participated in the Senseval 2 "*English all words*" task, presented in [Litkowski, 2001].

5.2 Performance of the Methods

Table 2 presents the accuracy of the six WSD methods, on the three files of Senseval 2. The presented accuracy corresponds to full coverage, and hence recall and precision are both equal to accuracy. The results in Table 2 suggest that our method outperforms that of Veronis and Ide, the hybrid method, and the random baseline. Moreover, our method achieved higher accuracy than the best unsupervised method that participated in Senseval 2, and overall slightly lower accuracy than the reported results of [Mihalcea *et al.*, 2004].

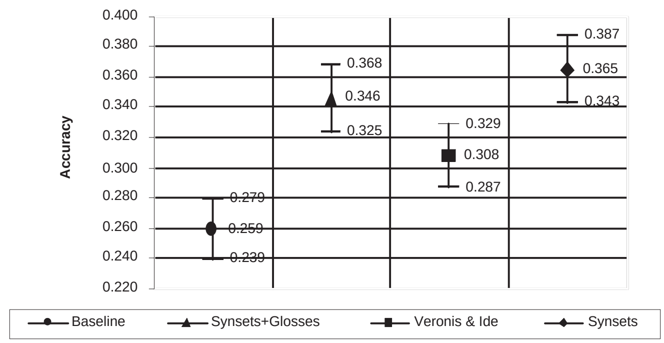

Figure 3: Accuracy on polysemous words and the respective 0.95 confidence intervals.

Figure 3 shows the corresponding overall results for the four methods we implemented, when accuracy is computed only on polysemous words, i.e. excluding trivial cases, along with the corresponding 0.95 confidence intervals. There is clearly a statistically significant advantage of our method (Synsets) over both the baseline and the method of Veronis and Ide. Adding WordNet's glosses to our method (Synsets+Glosses) does not lead to statistically significant difference (overlapping confidence intervals), and hence our method without glosses is better, since it is simpler and requires lower computational cost, as shown in section 5.3. The decrease in performance when adding glosses is justified by the fact that many of the glosses' words are not relevant to the senses the glosses express, and thus the use of glosses introduces irrelevant links to the SANs.

Figure 3 does not show the corresponding results of Mihalcea et al.'s method, due to the lack of corresponding published results; the same applies to the best unsupervised method of Senseval 2. We note that in the results presented by Mihalcea et al., there is no allusion to the variance in the accuracy of their method, which occurs by random assignment of senses to words that could not be disambiguated, nor to the number of these words. Thus no direct and clear statement can be made regarding their reported accuracy. In Figure 4 we

[1] In equation 3, $F_j \cdot A_j(p)$ is bounded and, hence, as p increases, eventually all nodes become inactive.

[2] If there is more than one sense node with this property per word, we select randomly. This never occurred in our experiments.

	Words		SAN	SAN Glosses	SAN	Baseline	Best Unsup.	Pagerank
	Mono	Poly	Synsets	Veronis and Ide	Synsets+Glosses		Senseval 2	Mihalcea
File 1 (d00)	**103**	**552**	**0.4595**	**0.4076**	**0.4396**	**0.3651**	unavailable	**0.4394**
File 2 (d01)	**232**	**724**	**0.4686**	**0.4592**	**0.4801**	**0.4211**	unavailable	**0.5446**
File 3 (d02)	**129**	**563**	**0.5578**	**0.4682**	**0.5115**	**0.4303**	unavailable	**0.5428**
Overall	**464**	**1839**	**0.4928**	**0.4472**	**0.4780**	**0.4079**	**0.4510**	**0.5089**

Table 2: Overall and per file accuracy on the Senseval 2 data set.

compare the accuracy of our method against Mihalcea et al.'s on each Senseval 2 file. In this case we included all words, monosemous and polysemous, because we do not have results for Mihalcea et al.'s method on polysemous words only; the reader should keep in mind that these results are less informative than the ones of Figure 3, because they do not exclude monosemous words. There is an overlap between the two

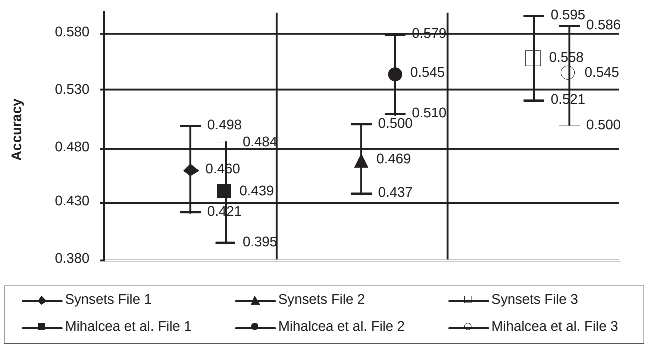

Figure 4: Accuracy on all words and the respective 0.95 confidence intervals.

confidence intervals for 2 out of 3 files, and thus the difference is not always statistically significant.

Regarding the best unsupervised method that participated in Senseval 2, we do not have any further information apart from its overall accuracy, and therefore we rest on our advantage in accuracy reported in Table 2. Finally, we note that to evaluate the significance of our weighting, we also executed experiments without taking it into account in the WSD process. The accuracy in this case drops by almost 1%, and the difference in accuracy between the resulting version of our method and the method of Veronis and Ide is no longer statistically significant, which illustrates the importance of our weighting. We have also conducted experiments in Senseval 3, where similar results with statistically significant differences were obtained: our method achieved an overall accuracy of 46% while Ide and Veronis achieved 39,7%. Space does not allow further discussion of our Senseval 3 experiments.

5.3 Complexity and Actual Computational Cost

Let k be the maximum branching factor (maximum number of edges per node) in a word thesaurus, l the maximum path length, following any type of semantic link, between any two nodes, and n the number of words to be disambiguated. Since we use breadth-first search, the computational complexity of constructing each SAN (network) is $O(n \cdot k^{l+1})$. Furthermore, considering the analysis of constrained spreading activation in [Rocha *et al.*, 2004], the computational complexity of spreading the activation is $O(n^2 \cdot k^{2l+3})$. The same computational complexity figures apply to the method of Veronis and Ide, as well as to the hybrid one, although k and l differ across the three methods. These figures, however, are worst case estimates, and in practice we measured much lower computational cost. In order to make the comparison of these three methods more concrete with respect to their actual computational cost, Table 3 shows the average numbers of nodes, edges, and iterations per network (sentence) for each method. Moreover, the average CPU time per network is shown (in seconds), which includes both network construction and activation spreading. The average time for the SAN Synsets method to disambiguate a word was 1.37 seconds. Table 3

	SAN Synsets	SAN Glosses Veronis and Ide	SAN Synsets+Glosses
Nodes/Net.	10,643.74	6,575.13	9,406.04
Edges/Net.	13,164.84	34,665.53	37,181.64
Pulses/Net.	166.93	28.64	119.15
Sec./Net.	13.21	3.35	19.71

Table 3: Average actual computational cost.

shows that our method requires less CPU time than the hybrid method, with which there is no statistically significant difference in accuracy; hence, adding glosses to our method clearly has no advantage. The method of Veronis and Ide has lower computational cost, but this comes at the expense of a statistically significant deterioration in performance, as discussed in Section 5.2. Mihalcea et al. provide no comparable measurements, and thus we cannot compare against them; the same applies to the best unsupervised method of Senseval 2.

6 Related Work

The majority of the WSD approaches proposed in the past deal only with nouns, ignoring other parts of speech. Some of those approaches [Yarowsky, 1995; Leacock *et al.*, 1998; Rigau *et al.*, 1997] concentrate on a set of a few pre-selected words, and in many cases perform supervised learning. In contrast, our algorithm requires no training, nor hand-tagged or parallel corpora, and it can disambiguate all the words in a text, sentence by sentence. Though WordNet was used in our experiments, the method can also be applied using LDOCE or Roget's thesaurus. Kozima and Furugori [1993] provide a straightforward way of extracting and using the semantic relations in LDOCE, while Morris and Hirst [1991] present a method to extract semantic relations between words from Roget's thesaurus.

7 Conclusions

We have presented a new unsupervised WSD algorithm, which utilizes all types of semantic relations in a word thesaurus. The algorithm uses Spreading Activation Networks (SANs), but unlike previous WSD work it creates SANs taking into account all sense-to-sense relations, rather than relations between senses and glosses, and it employs a novel edge-weighting scheme. The algorithm was evaluated on Senseval 2 data, using WordNet as the thesaurus, though it is general enough to exploit other word thesauri as well. It outperformed: (i) the most recent SAN-based WSD method, which overcame the problems older approaches faced, and (ii) the best unsupervised WSD method that participated in Senseval 2. It also matched the best WSD results that have been reported on the same data.

References

[Banerjee and Pedersen, 2003] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proc. of IJCAI-03*, pages 805–810, Acapulco, Mexico, 2003.

[Berger *et al.*, 2004] H. Berger, M. Dittenbach, and D. Merkl. An adaptive information retrieval system based on associative networks. In *Proc. of the 1st Asia-Pacific Conference on Conceptual Modelling*, pages 27–36, Dunedin, New Zealand, 2004.

[Bookman, 1987] L. Bookman. A microfeature based scheme for modelling semantics. In *Proc. of IJCAI-87*, pages 611–614, Milan, Italy, 1987.

[Cotrell and Small, 1983] G. Cotrell and S. Small. A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*, 6:89–120, 1983.

[Crestani, 1997] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11:453–482, 1997.

[Fellbaum, 1998] C. Fellbaum. *WordNet – an electronic lexical database*. MIT Press, 1998.

[Ide and Veronis, 1998] N. M. Ide and J. Veronis. Word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40, 1998.

[Kozima and Furugori, 1993] H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *Proc. of EACL-93*, pages 232–239, Utrecht, The Netherlands, 1993.

[Leacock *et al.*, 1998] C. Leacock, M. Chodorow, and G. A. Miller. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.

[Litkowski, 2001] K. Litkowski. Use of machine readable dictionaries for word-sense disambiguation in senseval-2. In *Senseval-2*, pages 107–110, Toulouse, France, 2001.

[Mavroeidis *et al.*, 2005] D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word sense for exploiting hierarchical thesauri in text classification. In *Proc. of PKDD-05*, pages 181–192, Porto, Portugal, 2005.

[Mihalcea *et al.*, 2004] R. Mihalcea, P. Tarau, and E. Figa. PageRank on semantic networks, with application to word sense disambiguation. In *Proc. of COLING-04*, Geneva, Switzerland, 2004.

[Moldovan and Rus, 2001] D. Moldovan and V. Rus. Explaining answers with Extended WordNet. In *Proc. of ACL-01*, Toulouse, France, 2001.

[Montoyo *et al.*, 2005] A. Montoyo, A. Suarez, G. Rigau, and M. Palomar. Combining knowledge- and corpus-based word sense disambiguation methods. *Journal of Artificial Intelligence Research*, 23:299–330, 2005.

[Morris and Hirst, 1991] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguistics*, 17:21–48, 1991.

[Palmer *et al.*, 2001] M. Palmer, C. Fellbaum, and S. Cotton. English tasks: All-words and verb lexical sample. In *Senseval-2*, pages 21–24, Toulouse, France, 2001.

[Patwardhan *et al.*, 2003] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proc. of CICLING-03*, pages 241–257, Mexico City, Mexico, 2003.

[Quillian, 1969] R. M. Quillian. The teachable language comprehender: a simulation program and theory of language. *Communications of ACM*, 12(8):459–476, 1969.

[Rigau *et al.*, 1997] G. Rigau, J. Atserias, and E. Agirre. Combining unsupervised lexical knowledge methods for word sense disambiguation. In *Proc. of ACL/EACL-97*, pages 48–55, Madrid, Spain, 1997.

[Rocha *et al.*, 2004] C. Rocha, D. Schwabe, and M. Poggi de Aragao. A hybrid approach for searching in the Semantic Web. In *Proc. of WWW-04*, pages 374–383, New York, NY, 2004.

[Sussna, 1993] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *2nd International Conference on Information and Knowledge Management*, pages 67–74, November 1993.

[Veronis and Ide, 1990] J. Veronis and N. M. Ide. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proc. of COLING-90*, pages 389–394, Helsinki, Finland, 1990.

[Voorhees, 1993] E. M. Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *Proc. of ACM SIGIR-93*, pages 171–180, Pittsburgh, PA, 1993.

[Yarowsky, 1995] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of ACL-95*, pages 189–196, Cambridge, MA, 1995.