

# PageRank on Semantic Networks, with Application to Word Sense Disambiguation

Rada Mihalcea, Paul Tarau, Elizabeth Figa

University of North Texas

Dallas, TX, USA

rada@cs.unt.edu, tarau@unt.edu, efiga@unt.edu

## Abstract

This paper presents a new open text word sense disambiguation method that combines the use of logical inferences with PageRank-style algorithms applied on graphs extracted from natural language documents. We evaluate the accuracy of the proposed algorithm on several sense-annotated texts, and show that it consistently outperforms the accuracy of other previously proposed knowledge-based word sense disambiguation methods. We also explore and evaluate methods that combine several open-text word sense disambiguation algorithms.

## 1 Introduction

Google's PageRank link-analysis algorithm (Brin and Page, 1998), and variants like Kleinberg's HITS algorithm (Kleinberg, 1999), have been used for analyzing the link-structure of the World Wide Web to provide global, content independent ranking of Web pages. Arguably, PageRank can be singled out as a key element of the paradigm-shift Google has triggered in the field of Web search technology, by providing a Web page ranking mechanism that relies on the collective knowledge of Web architects rather than content analysis of individual Web pages. In short, PageRank is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information. Applying a similar line of thinking to lexical and semantic knowledge graphs like WordNet (Miller, 1995) suggests using the implicit knowledge incorporated in their link structure for language processing applications, where knowledge drawn from an entire text can be used in making local ranking/selection decisions.

In this paper, we explore the applicability of PageRank to semantic networks, and show that such graph-based ranking algorithms can be successfully used in language processing applications. In particular, we propose and experiment with a new unsupervised knowledge-based word sense disambiguation algorithm, which succeeds in identifying the

sense of all words in open text with a precision significantly higher than other previously proposed knowledge-based algorithms.

The paper is organized as follows. Section 2 reviews the problem of word sense disambiguation, and surveys related work. Section 3 briefly describes the PageRank algorithm, and shows how this algorithm can be adapted to the WordNet graph. Section 4 introduces the PageRank-based word sense disambiguation algorithm. Combinations with other known algorithms are explored in Section 5. A thorough empirical evaluation of the proposed algorithms on several sense-annotated texts is provided in section 6.

## 2 Open Text Word Sense Disambiguation

The task of word sense disambiguation consists of assigning the most appropriate meaning to a polysemous word within a given context. Applications such as machine translation, knowledge acquisition, common sense reasoning, and others, require knowledge about word meanings, and word sense disambiguation is considered essential for all these applications.

Most of the efforts in solving this problem were concentrated so far toward targeted supervised learning, where each sense tagged occurrence of a particular word is transformed into a feature vector, which is then used in an automatic learning process. The applicability of such supervised algorithms is however limited only to those few words for which sense tagged data is available, and their accuracy is strongly connected to the amount of labeled data available at hand.

Instead, open-text knowledge-based approaches have received significantly less attention<sup>1</sup>. While the performance of such methods is usually exceeded by their supervised corpus-based alternatives, they have however the advantage of providing larger coverage.

---

<sup>1</sup>We use the term *knowledge-based* to denote methods that involve logical inferences and derivation of global properties that extend the data in a dictionary and/or a corpus with new knowledge. In our definition of *knowledge-based* approaches, the use of a corpus is not excluded.

Knowledge-based methods for word sense disambiguation are usually applicable to *all words* in open text, while supervised corpus-based techniques target only few selected words for which large corpora are made available. Four main types of knowledge-based methods have been developed so far for word sense disambiguation.

**Lesk algorithms.** First introduced by (Lesk, 1986), these algorithms attempt to identify the most likely meanings for the words in a given context based on a measure of contextual overlap between the dictionary definitions of the ambiguous words, or between the current context and dictionary definitions provided for a given target word.

**Semantic similarity.** Measures of semantic similarity computed on semantic networks (Rada et al., 1989). Depending on the size of the context they span, these measures are in turn divided into two main categories:

(1) Local context – where the semantic measures are used to disambiguate words additionally connected by syntactic relations (Stetina et al., 1998).

(2) Global context – where the semantic measures are employed to derive lexical chains, which are threads of meaning often drawn throughout an entire text (Morris and Hirst, 1991).

**Selectional preferences.** Automatically or semi-automatically acquired selectional preferences, as means for constraining the number of possible senses that a word might have, based on the relation it has with other words in context (Resnik, 1997).

**Heuristic-based methods.** These methods consist of simple rules that can reliably assign a sense to certain word categories: one sense per collocation (Yarowsky, 1993), and one sense per discourse (Gale et al., 1992).

In this paper, we propose a new open-text disambiguation algorithm that combines information drawn from a semantic network (WordNet) with graph-based ranking algorithms (PageRank). We compare our method with other open-text word sense disambiguation algorithms, and show that the accuracy achieved through our new PageRank-based method exceeds the performance obtained by other knowledge-based methods.

### 3 PageRank on Semantic Networks

In this section, we briefly describe PageRank (Brin and Page, 1998), and describe the view of WordNet as a graph, which facilitates the application of the graph-based ranking algorithm on this semantic network.

#### 3.1 The PageRank Algorithm

Iterative graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph; in the context of search engines, it is a way of deciding how important a page is on the Web. In this model, when one vertex links to another one, it is casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

Let  $G = (V, E)$  be a directed graph with the set of vertices  $V$  and set of edges  $E$ , where  $E$  is a subset of  $V \times V$ . For a given vertex  $V_i$ , let  $In(V_i)$  be the set of vertices that point to it, and let  $Out(V_i)$  be the set of edges going out of vertex  $V_i$ . The PageRank score of vertex  $V_i$  is defined as follows:

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

where  $d$  is a damping factor that can be set between 0 and 1<sup>2</sup>.

Starting from arbitrary values assigned to each node in the graph, the PageRank computation iterates until convergence below a given threshold is achieved. After running the algorithm, a fast in-place sorting algorithm is applied to the ranked graph vertices to sort them in decreasing order.

PageRank can be also applied on undirected graphs, in which case the out-degree of a vertex is equal to the in-degree of the vertex, and convergence is usually achieved after a fewer number of iterations.

#### 3.2 WordNet as a Graph

WordNet is a lexical knowledge base for English that defines words, meanings, and relations between them. The basic unit in WordNet is a synset, which is a set of synonym words or word phrases, and represents a concept. WordNet defines several semantic relations between synsets, including ISA relations (hypernym/hyponym), PART-OF relations (meronym/holonym), entailment, and others.

To represent WordNet as a graph, we use an *instance-centric* data representation, which defines

---

<sup>2</sup>The role of the damping factor  $d$  is to incorporate into the PageRank model the probability of jumping from a given vertex to another random vertex in the graph. In the context of Web surfing, PageRank implements the “random surfer model”, where a user clicks on links at random with a probability  $d$ , and jumps to a completely new page with probability  $1 - d$ . The factor  $d$  is usually set at 0.85 (Brin and Page, 1998), and this is the value we are also using in our implementation.

synsets as *vertices*, and relations or sets of relations as *edges*. The graph can be constructed as an undirected graph, with no orientation defined for edges, or as a directed graph, in which case a direction is arbitrarily established for each relation (e.g. hyponym  $\rightarrow$  hypernym).

Given a subset of the WordNet synsets, as identified in a given text or by other selectional criteria, and given a semantic relation, a graph is constructed by identifying all the synsets (*vertices*) in the given subset that can be linked by the given relation (*edges*). Relations can be also combined, for instance a graph can be constructed so that it accounts for both the ISA and the PART-OF relations between the vertices in the graph.

## 4 PageRank-based Word Sense Disambiguation

In this section, we describe a new unsupervised open-text word sense disambiguation algorithm that relies on PageRank-style algorithms applied on semantic networks.

### 4.1 Building the Text Synset Graph

To enable the application of PageRank-style algorithms to the disambiguation of all words in open text, we have to build a graph that represents the text and interconnects the words with meaningful relations.

Since no a-priori semantic information is available for the words in the text, we start with the assumption that every possible sense of a word is a potentially correct sense, and therefore all senses for all words are to be included in the initial search set. The synsets pertaining to all word senses form therefore the *vertices* of the graph. The *edges* between the nodes are drawn using synset relations available in WordNet, either explicitly encoded in the network, or derived by various means (see Sections 4.2, 4.3).

Note that not all WordNet arcs are suitable for combination with PageRank, as they sometimes identify competing word senses which tend to share targets of incoming or outgoing links. As our objective is to differentiate between senses, we want to focus on specific rather than shared links. We call two synsets *colexical* if they represent two senses of the same word – that is, if they share one identical lexical unit. For a given word or word phrase, colexical synsets will be listed as competing senses, from which a given disambiguation algorithm should select one.

To ensure that colexical synsets do not “contaminate” each other’s PageRank values, we have to make sure that they are not linked together, and hence they compete through disjoint sets of links. This means that relations between synsets pertaining

to various senses of the same word or word phrase are not added to the graph. Consider for instance the verb *travel*: it has six senses defined in WordNet, with senses 2 and 3 linked by an ISA relation (*travel#2* ISA *travel#3*). Since the synsets pertaining to these two senses are colexical (they share the lexical unit *travel*), this ISA link is not added to the text graph.

### 4.2 Basic Semantic Relations

WordNet explicitly encodes a set of basic semantic relations, including hypernymy, hyponymy, meronymy, holonymy, entailment, causality, attribute, pertainymy. WordNet 2.0 has also introduced nominalizations – which link verbs and nouns pertaining to the same semantic class, and domain links – a first step toward the classification of synsets, based on the “ontology” in which a given synset is relevant to. While the domain relations usually add a small number of links, their use tends to help focusing on a dominant field which was observed to help the disambiguation process.

### 4.3 Derived Semantic Relations

Two or more basic WordNet relations can be combined together to form a new relation. For instance, we can combine hypernymy and hyponymy to obtain the *coordinate* relation – which identifies synsets that share the same hypernym. For example, *dog#1* and *wolf#1* are *coordinates*, since they share the same hypernym *canine#1*.

It is worth mentioning the composite relation *xlink*, which is a new global relation that we define, which integrates all the basic relations (nominalizations and domain links included) and the coordinate relation. Shortly, two synsets are connected by an *xlink* relation if *any* WordNet-defined relation or a coordinate relation can be identified between them.

### 4.4 The PageRank Disambiguation Algorithm

The input to the disambiguation algorithm consists of raw text. The output is a text with word meaning annotations for all open-class words. Given a semantic relation *SR*, which can be a basic or composite relation, the algorithm consists of the following main steps:

#### Step 1: Preprocessing.

During preprocessing, the text is tokenized and annotated with parts of speech. Collocations are identified using a sliding window approach, where a collocation is considered to be a sequence of words that forms a compound concept defined in WordNet. Named entities are also identified at this stage.

#### Step 2: Graph construction.

Build the text synset graph: for all open class words in the text, identify all synsets defined in WordNet, and add them as vertices in the graph. Words

previously assigned with a named entity tag, and modal/auxiliary verbs are not considered. For the given semantic relation  $SR$ , add an edge between all vertices in the graph that can be linked by the relation  $SR$ .

### Step 3: PageRank.

Assign an initial small value to each vertex in the graph. Iterate the PageRank computation until it converges - usually for 25-30 iterations. In our implementation, vertices are initially assigned with a value of 1. Notice that the final values obtained after PageRank runs to completion are not affected by the choice of the initial value, only the number of iterations to convergence may be different.

### Step 4: Assign word meanings.

For each ambiguous word in the text, find the synset that has the highest PageRank score, which is uniquely identifying the sense of the word. If none of the synsets corresponding to the meanings of a word could be connected with other synsets in the graph using the given relation  $SR$ , the word is assigned with a random sense (when the WordNet sense order is not considered), or with the first sense in WordNet (when a sense order is available).

The algorithm can be run on the entire text at once, in which case the resulting graph is fairly large - usually more than two thousands vertices - and has high connectivity. Alternatively, it can be run on smaller sections of the text, and in this case the graphs have lower number of vertices and lower connectivity. In the experiments reported in this paper, we are using the first option, since it results in richer synset graphs and ensures that most of the words are assigned a meaning using the PageRank sense disambiguation algorithm.

## 5 Related Algorithms

We overview in this section two other word sense disambiguation algorithms that address all words in open text: Lesk algorithm, and the most frequent sense algorithm<sup>3</sup>. We also propose two new hybrid algorithms that combine the PageRank word sense disambiguation method with the Lesk algorithm and the most frequent sense algorithm.

### 5.1 The Lesk algorithm

The Lesk algorithm (Lesk, 1986) is one of the first algorithms used for the semantic disambiguation of all words in open text. The only resource required by the algorithm is a set of dictionary entries, one for each possible word sense, and knowledge about the immediate context where the sense disambiguation is performed.

<sup>3</sup>The reason for choosing these algorithms over the other methods mentioned in section 2 is the fact that they address *all* open class words in a text.

The main idea behind the original definition of the algorithm is to disambiguate words by finding the overlap among their sense definitions. Namely, given two words,  $W_1$  and  $W_2$ , each with  $N_{W_1}$  and  $N_{W_2}$  senses defined in a dictionary, for each possible sense pair  $W_1^i$  and  $W_2^j$ ,  $i=1..N_{W_1}$ ,  $j=1..N_{W_2}$ , first determine their definitions overlap, by counting the number of words they have in common. Next, the sense pair with the highest overlap is selected, and consequently a sense is assigned to each of the two words involved in the initial pair.

When applied to open text, the original definition of the algorithm faces an explosion of word sense combinations<sup>4</sup>, and alternative solutions are required. One solution is to use simulated annealing, as proposed in (Cowie et al., 1992). Another solution - which we adopt in our experiments - is to use a variation of the Lesk algorithm (Kilgarriff and Rosenzweig, 2000), where meanings of words in the text are determined individually, by finding the highest overlap between the sense definitions of each word and the current context. Rather than seeking to simultaneously determine the meanings of all words in a given text, this approach determines word senses individually, and therefore it avoids the combinatorial explosion of senses.

### 5.2 Most Frequent Sense

WordNet keeps track of the frequency of each word meaning within a sense-annotated corpus. This introduces an additional knowledge-element that can significantly improve the disambiguation performance.

A very simple algorithm that relies on this information consists of picking the most frequent sense for any given word as the correct one. Given that sense frequency distributions tend to decrease exponentially for less frequent senses, this guess usually outperforms methods that use exclusively the content of the document and associated dictionary information.

### 5.3 Combining PageRank and Lesk

When combining two different algorithms, we have to ensure that their effects accumulate without disturbing each algorithms internal workings.

The PageRank+Lesk algorithm consists in providing a default ordering by Lesk (possibly after shuffling WordNet senses to remove the sense frequency bias), and then applying PageRank, which

<sup>4</sup>Consider for instance the text "I saw a man who is 108 years old and can still walk and tell jokes", with nine open class words, each with several possible senses : see(26), man(11), year(4), old(8), can(5), still(4), walk(10), tell(8), joke(3). Given the total of 43,929,600 possible sense combinations, finding the optimal combination using definition overlaps is not a tractable approach.

	Size(words)	Random	Lesk	PageRank	PageRank+Lesk
<b>SEMCOR</b>					
law	825	37.12%	39.62%	46.42%	49.36%
sports	808	29.95%	33.00%	40.59%	46.18%
education	898	37.63%	41.33%	46.88%	52.00%
debates	799	40.17%	42.38%	47.80%	50.52%
entertainment	802	39.27%	43.05%	43.89%	49.31%
AVERAGE	826	36.82%	39.87%	45.11%	49.47%
<b>SENSEVAL-2</b>					
d00	471	28.97%	43.94%	43.94%	47.77%
d01	784	45.47%	52.65%	54.46%	57.39%
d02	514	39.24%	49.61%	54.28%	56.42%
AVERAGE	590	37.89%	48.73%	50.89%	53.86%
AVERAGE (ALL)	740	37.22%	43.19%	47.27%	51.16%

Table 1: Word Sense Disambiguation accuracy for PageRank, Lesk, PageRank+Lesk, and Random (no sense order)

will eventually reorder the senses. With this approach, senses that have similar PageRank values will keep their Lesk ordering. As PageRank overrides Lesk one can notice that in this case we prioritize PageRank, which tends to outperform Lesk. The resulting algorithm provides a combination which improves over both algorithms individually, as shown in Section 6.

#### 5.4 Combining PageRank with the Sense Frequency

The combination of PageRank with the WordNet sense frequency information is done in two steps:

- introduce the WordNet frequency ordering by removing the random permutation of senses
- use a formula which combines PageRank and actual WordNet sense frequency information

While a simple product of the two ranks already provides an improvement over both algorithms the following formula which prioritizes the first sense provides the best results:

$$Rank = \begin{cases} 4 \times FR \times PR & \text{if } N = 1 \\ FR \times PR & \text{if } N > 1 \end{cases}$$

where  $FR$  represents the WordNet sense frequency,  $PR$  represents the rank computed by PageRank,  $N$  is the position in the frequency ordered synset list, and  $Rank$  represents the combined rank.

## 6 Experimental Evaluation

We evaluate the accuracy of the word sense disambiguation algorithms on a benchmark of sense-annotated texts, in which each open-class word is mapped to the meaning selected by a lexicographer as being the most appropriate one in the context of a sentence. We are using a subset of the SemCor texts (Miller et al., 1993) – five randomly selected files covering different topics: news, sports, entertainment, law, and debates – as well as the data

set provided for the English all words task during SENSEVAL-2.

The average size of a file is 600-800 open class words. On each file, we run two sets of evaluations. (1) One set consisting of the basic “uninformed” version of the knowledge-based algorithms, where the sense ordering provided by the dictionary is not taken into account at any point. (2) A second set of experiments consisting of “informed” disambiguation algorithms, which incorporate the sense order rendered by the dictionary.

### 6.1 Uninformed Algorithms

Given that word senses are ordered in WordNet by decreasing frequency of their occurrence in large sense annotated data, we explicitly remove this ordering by applying a random permutation of the senses with uniform distribution. This randomization step ensures that any eventual bias introduced by the sense ordering is removed, and it enables us to evaluate the impact of the disambiguation algorithm when no information about sense frequency is available. In this setting, the following dictionary-based algorithms are evaluated and compared: PageRank, Lesk, combined PageRank-Lesk, and the random baseline:

**PageRank.** The algorithm introduced in this paper, which selects the most likely sense of a word based on the PageRank score assigned to the synsets corresponding to the given word within the text graph. While experiments were performed using all semantic relations listed in Sections 4.2 and 4.3, we report here on the results obtained with the *xlink* relation, which was found to perform best as compared to other semantic relations.

**Lesk.** We are also experimenting with the Lesk algorithm described in section 5.1, which decides on the correct sense of a word based on the highest

	Size(words)	MFS	Lesk	PageRank	PageRank+Lesk
<b>SEMCOR</b>					
law	825	69.09%	72.65%	73.21%	73.97%
sports	808	57.30%	64.21%	68.31%	68.31%
education	898	64.03%	69.33%	71.65%	71.53%
debates	799	66.33%	70.07%	71.14%	71.67%
entertainment	802	59.72%	64.98%	66.02%	66.16%
AVERAGE	826	63.24%	68.24%	70.06%	70.32%
<b>SENSEVAL-2</b>					
d00	471	51.70%	53.07%	58.17%	57.74%
d01	784	60.80%	64.28%	67.85%	68.11%
d02	514	55.97%	62.84%	63.81%	64.39%
AVERAGE	590	56.15%	60.06%	63.27%	63.41%
AVERAGE (ALL)	740	60.58%	65.17%	67.51%	67.72%

Table 2: Word Sense Disambiguation accuracy for PageRank, Lesk, PageRank+Lesk, and Most Frequent Sense (WordNet sense order integrated)

overlap between the dictionary sense definitions and the context where the word occurs.

**PageRank + Lesk.** The PageRank and Lesk algorithms can be combined into one hybrid algorithm, as described in section 5.3. First, we order the senses based on the score assigned by the the Lesk algorithm, and then apply PageRank on this reordered set of senses.

**Random.** Finally, we are running a very simple sense annotation algorithm, which assigns a random sense to each word in the text, and which represents a baseline for this set of “uninformed” word sense disambiguation algorithms.

Table 1 lists the disambiguation precision obtained by each of these algorithms on the evaluation benchmark. On average, PageRank gives an accuracy of 47.27%, which brings a significant 7.7% error reduction with respect to the Lesk algorithm, and 19.0% error reduction over the random baseline. The best performance is achieved by a combined PageRank and Lesk algorithm: 51.16% accuracy, which brings a 28.5% error reduction with respect to the random baseline. Notice that all these algorithms rely exclusively on information drawn from dictionaries, and do not require any information on sense frequency, which makes them highly portable to other languages.

## 6.2 Informed Algorithms

In a second set of experiments, we allow the disambiguation algorithms to incorporate the sense order provided by WordNet. While this class of algorithms is *informed* by the use of global frequency information, it does not use any specific corpus annotations and therefore it leans in gray area between supervised and unsupervised methods. We are again evaluating four different algorithms: PageRank, Lesk, combined PageRank – Lesk, and a

baseline consisting of assigning by default the most frequent sense.

**PageRank.** The PageRank-based algorithm introduced in this paper, combined with the WordNet sense frequency, as described in Section 5.4.

**Lesk.** The Lesk algorithm described in section 5.1, applied on an ordered set of senses. This means that words that have two or more senses with a similar score identified by Lesk, will keep the WordNet sense ordering.

**PageRank + Lesk.** A hybrid algorithm, that combines PageRank, Lesk, and the dictionary sense order. This algorithm consists of the method described in Section 5.3, applied on the ordered set of senses.

**Most frequent sense.** Finally, we are running a simple “informed” sense annotation algorithm, which assigns by default the most frequent sense to each word in the text (i.e. sense number one in WordNet).

Table 2 lists the accuracy obtained by each of these informed algorithms on the same benchmark. Again, the PageRank algorithm exceeds the other knowledge-based algorithms by a significant margin: it brings an error rate reduction of 21.3% with respect to the most frequent sense baseline, and a 7.2% error reduction over the Lesk algorithm. Interestingly, combining PageRank and Lesk under this informed setting does not bring any significant improvements over the individual algorithms: 67.72% obtained by the combined algorithm compared with 67.51% obtained with PageRank only.

## 6.3 Discussion

Regardless of the setting – fully unsupervised algorithms with no a-priori knowledge about sense order, or informed methods where the sense order rendered by the dictionary is taken into account – the PageRank-based word sense disambiguation algorithm exceeds the baseline by a large margin, and

always outperforms the Lesk algorithm. Moreover, a hybrid algorithm that combines the PageRank and Lesk methods into one single algorithm is found to improve over the individual algorithms in the first setting, but brings no significant changes when the sense frequency is also integrated into the disambiguation algorithm. This may be explained by the fact that the additional knowledge element introduced by the sense order in WordNet increases the redundancy of information in these two algorithms to the point where their combination cannot improve over the individual algorithms.

The most closely related method is perhaps the lexical chains algorithm (Morris and Hirst, 1991) – where threads of meaning are identified throughout a text. Lexical chains however only take into account possible relations between concepts in a static way, without considering the importance of the concepts that participate in a relation, which is recursively determined by PageRank. Another related line of work is the word sense disambiguation algorithm proposed in (Veronis and Ide, 1990), where a large neural network is built by relating words through their dictionary definitions.

**The Analogy.** In the context of Web surfing, PageRank implements the “random surfer model”, where a user surfs the Web by following links from any given Web page. In the context of text meaning, PageRank implements the concept of text cohesion (Halliday and Hasan, 1976), where from a certain concept  $C$  in a text, we are likely to “follow” links to related concepts – that is, concepts that have a semantic relation with the current concept  $C$ .

Intuitively, PageRank-style algorithms work well for finding the meaning of all words in open text because they combine together information drawn from the entire text (*graph*), and try to identify those synsets (*vertices*) that are of highest importance for the text unity and understanding.

The meaning selected by PageRank from a set of possible meanings for a given word can be seen as the one *most recommended* by related meanings in the text, with preference given to the “recommendations” made by *most influential* ones, i.e. the ones that are in turn highly recommended by other related meanings. The underlying hypothesis is that in a cohesive text fragment, related meanings tend to occur together and form a “Web” of semantic connections that approximates the model humans build about a given context in the process of discourse understanding.

## 7 Conclusions

In this paper, we showed that iterative graph-based ranking algorithms – originally designed for content-independent Web link analysis or for social

networks – turn into a useful source of information for natural language tasks when applied on semantic networks. In particular, we proposed and evaluated a new approach for unsupervised knowledge-based word-sense disambiguation that relies on PageRank-style algorithms applied on a WordNet-based concepts graph, and showed that the accuracy achieved through our algorithm exceeds the performance obtained by other knowledge-based algorithms.

## Acknowledgments

This work was partially supported by a National Science Foundation grant IIS-0336793.

## References

- S. Brin and L. Page. 1998. The anatomy of a large-scale hyper-textual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.
- J. Cowie, L. Guthrie, and J. Guthrie. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the 5th International Conference on Computational Linguistics COLING-92*, pages 157–161.
- W. Gale, K. Church, and D. Yarowsky. 1992. One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Harriman, New York.
- M. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman.
- A. Kilgarriff and R. Rosenzweig. 2000. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34:15–48.
- J.M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.
- M.E. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference 1986*, Toronto, June.
- G. Miller, C. Leacock, T. Randee, and R. Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, New Jersey.
- G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.
- J. Morris and G. Hirst. 1991. Lexical cohesion, the thesaurus, and the structure of text. *Computational Linguistics*, 17(1):21–48.
- R. Rada, H. Mili, E. Bickell, and B. Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19:17–30, Jan/Feb.
- P. Resnik. 1997. Selectional preference and sense disambiguation. In *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?*, Washington DC, April.
- J. Stetina, S. Kurohashi, and M. Nagao. 1998. General word sense disambiguation method based on a full sentential context. In *Usage of WordNet in Natural Language Processing, Proceedings of COLING-ACL Workshop*, Montreal, Canada, July.
- J. Veronis and N. Ide. 1990. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING 1990)*, Helsinki, Finland, August.
- D. Yarowsky. 1993. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop*.