A Logic Programming Framework for Semantic Interpretation with WordNet and PageRank

Paul Tarau
Department of Computer Science and Engineering
University of North Texas
P.O. Box 311366
Denton, Texas 76203
E-mail: tarau@cs.unt.edu
WWW: http://www.cs.unt.edu/∼tarau

Rada Mihalcea
Department of Computer Science and Engineering
University of North Texas
P.O. Box 311366
Denton, Texas 76203
E-mail: rada@cs.unt.edu
WWW: http://www.cs.unt.edu/∼rada

Elizabeth Figa
School of Library and Information Sciences
University of North Texas
P.O. Box 311068
Denton, Texas 76203
E-mail: efiga@lis.admin.unt.edu
WWW: http://courses.unt.edu/efiga

Abstract

This paper describes applications of Logic Programming to Natural Language processing in combination with graph-algorithms and statistical methods. Google's PageRank and similar fast-converging recursive graph algorithms have provided practical means to statically rank vertices of large graphs like the World Wide Web. By combining a fast Java-based PageRank implementation with a Prolog base inferential layer, running on top of an optimized WordNet graph, we describe applications to word sense disambiguation and evaluate their accuracy in comparison with human annotated corpus data.

Keywords: Logic Programming based Natural Language Processing, word-sense disambiguation, PageRank-style graph algorithms, inferential extensions to WordNet, semantic information processing.

1 Introduction

Google's PageRank [18, 3] link-analysis algorithm and variants like Kleinberg's HITS algorithm [11] 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 Web pages. Applying a similar line of thinking to large lexical and semantic information graphs like WordNet [15, 4, 14] and FrameNet [1] suggests using the implicit knowledge incorporated in their link structure for tasks ranging from automated extraction of top-level ontologies to word sense disambiguation, document summarization and text-mining.

2 The WordNet Lexical Knowledge Base

In a long-term collaborative effort that began in 1985, an interdisciplinary group at Princeton developed WordNet as a "machine readable lexical database organized by meanings". WordNet is not a simple dictionary – dictionaries help to settle issues of word use or sense priority – but they do not address issues of synchronic organization of lexical knowledge as WordNet does [15].

WordNet divides the lexicon into five categories: *nouns, verbs, adjectives, adverbs* (which are connected by various semantic and lexical relations), as well as *functions words* (which are not included, as they are assumed part of the syntactic component of the language). Each of these lexical structures reflects a different way of categorizing experience. WordNet organizes lexical information in terms of word meanings rather than word forms in a way similar to a thesaurus.

WordNet maps word forms and word meanings as a many-to-many relation, which indicates that some forms have several different meanings, and some meanings can be expressed by different forms.

Several semantic relationships are covered by WordNet. For instance, a hyponym is a meaning that acquires all the features of its hypernym, which is a more generic concept; for example, oak is a hyponym of tree. Meronymy is the relation of being part of; for example, arm is a meronym of body.

These relations are defined in WordNet between *meanings* instead of being defined between *words or word phrases.*

Meanings are represented by *integers* called *synsets*, associated to sets of words and word phrases collectively defining a sense element (concept, predicate or property).

So, for example, the meaning identifier (synset) `Id=100011413` maps to the following list of words and word phrases: `[[animal], [animate,being], [beast], [brute], [creature], [fauna]]`, which collectively define a common *meaning.*

The WordNet database [15] is available in Prolog form (see `http://www.cogsci.princeton.edu/~wn`) and is therefore ready to be used as part of a rule-based inference system.

We have refactored the set of predicates provided by WordNet closely following the WordNet relation set (see `http://www.cogsci.princeton.edu/~wn/doc.shtml`) to support bidirectional constant time access to the set of *meanings* associated to a given word phrase (indexed by a unique head word) and for the set of word phrases and relations associated to a given (unique) meaning.

A reverse index going from words to the sysnsets in which they occur provides fast access from lexical froms to the list of their possible meanings, like in the following example:

```
w(accommodate,[81108,92819,85354,92990,81998,83880,92580]).
w(accommodating,[93322,99933]).
```

Definitions and examples originally present in WordNet glosses are preparsed so that they can be processed efficiently, if needed, at runtime. We also collect frequency information and word forms not present in the form of WordNet entries. Note also the presence of reversed relations like hyponyms (reverse hypernyms) and reverse meronyms. These are are precomputed to support high performance graph walk operations.

3 Building the WordNet Graph

We have converted the refactored Prolog WordNet database described in [6, 5] to a Java graph representation using Jinni 2004's [27] built-in graph processing libraries. We have chosen an *instance-centric* data representation which defines *vertices* as *synsets* and *synsets with lexical variant numbers* and edges as sets of relations. Given the comparatively small number of different semantic relations between such vertex instances, this results in improved indexing and keeps the inverted graph needed to compute PageRank at a manageable size. The memory footprint of the WordNet graph, including the data areas needed for the PageRank computation and the sorting by rank, fits in the main memory of a typical Windows PC, in a Java workspace with peak memory usage of around 320MBytes. The resulting graph is built from a Prolog representation in about 4 hours including parsing of the data files and computation of inverted edges[1].

4 Implementing the PageRank Algorithm

We will now shortly describe the PageRank algorithm, following [18, 3].

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 $V \times V$. We assume a vertex P has vertices $P_1 ... P_n$ which point to it. The parameter d is a damping factor which can be set between 0 and 1. Let $OUT(P)$ be the number of edges going out of vertex P. The PageRank of a vertex P_0 is given as follows:

$$PR(P_0) = (1 - d) + d * (PR(P_1)/OUT(P_1) + ... + PR(P_n)/OUT(P_n))$$

Note that somewhat faster or more compact (but significantly more complex) implementations have been recently suggested [9, 2, 22], although the additional time or memory savings are more relevant in cases like the complete Web graph than it would be in the case of moderately large graphs like our WordNet graph.

We have implemented a simple linear-time variant of the PageRank algorithm [18] which is now part of Jinni 2004's [27] graph processing API. Starting from arbitrary values assigned to each node, the code iterates the PR computation until convergence below a given threshold is noticed. It has been proven in [18] that the algorithm converges and we have noticed on a large sample of random graphs that this usually happens in less than 30 iterations.

After running the algorithm, a fast in-place *merge-sort* is a applied to the ranked graph vertices to sort them in decreasing order.

Given the tightly integrated Java and Prolog processing possible through Jinni 2004's reflection based Java interface [28] the resulting rank-sorted graph is available for Prolog processing as if they were prolog facts, through a simple API with operations like the following:

```
vertex_of(G,V):-
  vertex_iterator(G,Vs),
```

[1]The significant total execution time comes from the fact that, given that the refactoring process is a one time operation, we have implemented it in Prolog by using sorted lists as representations for sets of vertices and edges instead of side effects or external hash tables.

```
  iterator_element(Vs,V).

incoming_of(G,V,IV):-
  in_iterator(G,V,Vs),
  iterator_element(Vs,IV).

outgoing_of(G,V,OV):-
  out_iterator(G,V,Vs),
  iterator_element(Vs,OV).

run_page_rank(G,Times):-
  invoke_java_method(G,runPageRank,Times).

rank_sort(G):-invoke_java_method(G,rankSort,_).
```

We refer to [27] for a description of the underlying bidirectional Prolog-Java interface.

5 View From the Top: PageRanking the WordNet Graph

WordNet 2.0 contains 9 top-level nouns and a few hundred top-level verbs. The top-level nouns are the following:

```
[entity]
[psychological,feature]
[abstraction]
[state]
[event]
[act] [human,action] [human,activity]
[group] [grouping]
[possession]
[phenomenon]
```

After running PageRank on the WordNet graph, we found the following top ranked synsets (sets of word phrases with similar meanings):

```
rank :: synset given as a word phrase set
------------------------------------------------
3107 :: [entity]*
2366 :: [group] [grouping]*
2088 :: [object] [physical,object]
1716 :: [person] [individual]
        [someone] [somebody]
        [mortal] [human] [soul]
1532 :: [artifact] [artefact]
1494 :: [taxonomic,group]
        taxonomic,category] [taxon]
1301 :: [biological,group]
1238 :: [science] [scientific,discipline]
1159 :: [abstraction]*

1136 :: [natural,science]
1039 :: [act] [human,action] [human,activity]*
```

```
 984 :: [biology] [biological,science]
 976 :: [life,science] [bioscience]
 907 :: [cognition] [knowledge] [noesis]
------------------------------------------------
```

Unsurprisingly, nouns close to the top-level have the highest ranks. On the other hand, only 3 out of the 9 top-level WordNet categories (marked with a *) are present among the top 9 ranked concepts. The highest ranking of these concepts shows WordNet's implicit *voting* for a *top-level ontology* [10, 24] towards where most semantic links converge, in a way similar to popular Web sites towards which most other pages refer.

It is interesting to note that the top-ranked verbs express change, movement and interaction rather than existence, possession or situation.

```
rank :: synset given as a word phrase set

620  :: [change],
330  :: [change] [alter] modify],
262  :: [move],
224  :: [act] [move]
204  :: [move],[displace]
169  :: [interact]
150  :: [communicate] [intercommunicate]
```

Arguably, this points out that the dominant verb ontology implicit in WordNet is oriented toward describing change, movement and interaction. Not far away, among the top verbs we find, ranked 106, `[make] [create]`, which positions creation verbs (instances of change) as important.

6 Specializing PageRank to a Given Semantic Relation and a Given Document

WordNet provides some basic semantic relations like hypernymy/hyponymy and meronymy, as well as derived relations like coordination (when two concepts have the same hypernym). We can build the subgraph by selecting only a given relation and by ranking possible meanings of the document based on that.

A sketch of the algorithm looks as follows:

1. Read, tokenize and group a text in sentences.
2. Look-up the synsets associated to word phrases in each sentence.
3. Build a synset graph as follows:
   - create an empty graph
   - for each synset of each word phrase occurring in the text add a vertex
4. For a selected set of WordNet relations add to the graph each edge where the link points to
5. Apply PageRank and sort the vertices in decreasing rank order

In this case, by defining relations which discriminate between competing senses of a word or word phrase we will show that choosing the highest ranked synsets for each alternative meaning will provide an effective method to *disambiguate* the text.

If we chose the *coordinate term* relation, the highest ranked synsets are likely to represent the focus of the text and provide hints about dominant internal connections between sentences and as such they can be used to discover implicit information in the text or for summarization.

7 Towards Knowledge-based Word Sense Disambiguation

As in the case of a Web search, where PageRank is combined with content and meta-tag matching, we can use precomputed or dynamically computed PageRank values to rank possible senses of words (disambiguation).

We will first describe some variations of this general idea and then evaluate their performance on a set of standard disambiguation benchmarks.

7.1 Related work on word-sense disambiguation

Knowledge-based methods represent a distinct category in Word Sense Disambiguation (WSD). While the performance of such knowledge intensive methods is usually exceeded by their corpus-based alternatives (see for instance [30], [17]), they have however the advantage of providing larger coverage. Knowledge-based methods for WSD are usually applicable to *all words* in open text, while 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 WSD[2].

1. Lesk algorithms [12], where the word sense most likely to be chosen in a given context is decided based on a measure of contextual overlap between the current context and dictionary definitions provided for the target word.
2. Measures of semantic similarity computed on semantic networks [21]. This category includes methods that compute the semantic density/distance between concepts. Depending on the size of the context they span, the application of these measures to the problem of semantic disambiguation is in turn divided into two main categories:
   - Methods applicable to a local context, where the semantic measures are used to disambiguate words additionally connected by (a) syntactic relations [25]; or (b) their locality [19].
   - Methods applicable to the global context, where the semantic measures are employed to derive lexical chains, which are threads of meaning often drawn throughout an entire text [13].
3. 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 [23].
4. Heuristic based methods, which consist of simple rules that can reliably assign a sense to certain word categories.

[2]These techniques are generally applicable in conjunction with any knowledge base, even though most of them were applied and tested on WordNet

- One sense per collocation [29].
- One sense per discourse [7].

In this paper, we introduce a new approach to knowledge-based word-sense disambiguation that relies on PageRank-style algorithms applied on semantic networks. We compare our method with other dictionary-based algorithms (in particular, Lesk algorithms), and show that the accuracy achieved through our new PageRank method exceeds the performance obtained by other dictionary-based algorithms.

7.2 Sense Discriminating Relations

Most basic WordNet relations do not work well in combination with PageRank as they tend to identify competing word senses which tend to share targets of incoming or outgoing links.

We call two synsets *colexical* if they have at least one shared representation to the same word or word phrase. This means that 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 evaluate the accuracy of such algorithms, we will run them against a benchmark of human annotated files where each word phrase is mapped to the synset selected by a human lexicographer as being the most appropriate one in the context of a sentence.

To ensure that colexical synsets compete through disjoint sets of links and will not "contaminate" each other's Pagerank values, we will have to make sure that senses corresponding to an identical lexical representation as a word or word phrase are separated. This is achieved by filtering various WordNet relations with the predicate `colex/2` as in:

```
xlinked(I,J):-linked(I,J),not(colex(I,J)).
```

The relation `linked(I,J)` holds for values `I` and `J` such that it exists at least one WordNet link or reverse link between I and J.

The composite relation `xlinked(I,J)` show in the example, which is efficiently computed by our Prolog data representation, turned out to work well for most of the disambiguation tests.

7.3 The Disambiguation Algorithm

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 a 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 nodes of the graph. The arcs between the nodes are drawn using synset relations available in WordNet. See the previous section for sense discriminating relations extracted from WordNet.

Besides the link graph, two additional graphs, are built:

- The Word Phrase Graph which contains links to the synsets related to a word
- The Sentence Graph that contains links to word phrases occurring in the sentence

After the text graph is constructed, PageRank is applied to identify a score corresponding to each synset in the graph. Among all synsets corresponding to a given ambiguous word, the one that has the highest rank is selected, which is uniquely identifying the sense of the word. As Jinni 2004 provides hashing on ground terms and graph are implemented through hashing tables, moving betwen Synsets, Words and Sentecnes requires $O(1)$ effort.

7.4 Experimental Evaluation

To evaluate the performance of the word sense disambiguation algorithm, we use a subset of SemCor [16] – a fairly large textual corpus where words are tagged with their corresponding sense in WordNet. The texts in SemCor were extracted from the Brown Corpus and then linked to senses in WordNet. The tagging of SemCor was performed manually, and therefore this corpus can be considered a "gold-standard" for the evaluation of word sense disambiguation algorithms.

We randomly selected ten SemCor files, covering different topics (news, justice, sports, etc.), and used them in the disambiguation experiments. The average size of a file is 800 open class words (nouns, verbs, adjectives, adverbs) which are passed on to the disambiguation algorithm.

On each file, we run three sense annotation experiments.

- **PageRank.** This is the PageRank-based algorithm introduced in this paper, which selects the most likely sense of a word based on the rank assigned to the synsets corresponding to the given word within the text graph.
- **Lesk.** For comparative evaluations, we have also implemented the Lesk algorithm [12], which decides on the correct sense of a word based on the highest overlap measured between the dictionary sense definitions, and the context where the word occurs.
- **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 unsupervised word sense disambiguation.

Table 1 lists the disambiguation precision obtained by each of these algorithms on the SemCor subset.

Domain	Size(words)	Random	Lesk	PageRank
law	857	37.57%	44.34%	64.84%
law	777	38.00%	43.62%	61.78%
sports	781	34.18%	39.56%	52.22%
sports	861	35.77%	39.95%	48.73%
sports	780	36.28%	42.94%	50.39%
sports	770	32.07%	43.37%	56.42%
sports	781	35.46%	40.58%	54.34%
education	920	34.56%	47.17%	58.24%
war	839	34.92%	42.55%	60.57%
entertainment	868	42.16%	44.23%	54.86%
AVERAGE	823	36.09%	42.83%	56.23%

Table 1: Word Sense Disambiguation accuracy for PageRank, Lesk, and Random

On average, PageRank gives an accuracy of 56%, which brings a 23% error reduction with respect to the Lesk algorithm, and 32% error reduction over the random baseline. Notice that all these algorithms rely exclusively on information drawn from dictionaries, and do not require any sense annotated data, which makes them highly portable to other languages.

8 PageRank for Concept Extraction

Clearly, the highest ranked synsets already provide a suggestion for key concepts found in the document. However, it seems interesting to extend the link graph with *implicit* semantic information obtained through the Prolog inferential layer.

8.1 Filtering out Dominant Verbs

A rough approximation of the implicit WordNet ontology applied to sentences is obtained by interpreting verbs as predicates having nouns occurring in the text as their arguments. WordNet organizes noun synsets in inheritance trees and organizes verbs in a forest of shallow trees connected weekly by entailment and causality relations. As frequently occurring verbs are used repeatedly to connect nouns in sentences, in longer documents WordNet's top-level verbs will usually dominate the PageRanked graph like in the following example from the Brown corpus we have used in our disambiguation algorithm

```
synset=91272,rank=6.30,[[act],[move]])
synset=88639,rank=5.74,[[move]])
synset=87628,rank=5.42,[[make],[create]])
synset=88652,rank=4.67,[[travel],[go],[move],[locomote]])
synset=90463,rank=4.39,[[give]])
synset=90510,rank=4.26,[[get],[acquire]])
synset=81377,rank=3.60,[[end],[terminate]])
synset=83002,rank=3.42,[[see],[consider],[reckon],[view],[regard]])
```

If the intent of the analysis is to find what a document is about - with emphasis on noun phrase keywords and sentences about them, we will have to *reflect back* the high ranks of dominant verbs towards their noun arguments. Given that the key iterative step of PageRank consists in propagating values from nodes ranked in the previous step through their outgoing links, we will have to add to our graph links which connect in various ways these verbs to their arguments.

8.2 Extending the Link Graph by Definitions in the WordNet Glosses

A first technique consist in parsing the glosses (definitions and/or examples of use) present in WordNet and make their synsets (especially the ones coming from verbs) point towards the synsets that they contribute to define or to exemplify (especially nouns). The following Prolog code provides an example of such synset extraction from precompiled WordNet glosses:

```
% Op=definition or example
% T=syntactic type
% I=synset the gloss belongs to
% GI=synset of syntactic type T extracted from the gloss

gloss2i(Op,T,I,GI):-
  g(I,Sense), % glosses
```

```
member(X,Sense), % extract a def. or example X
  functor(X,Op,_),arg(1,X,Ts),
  member(W,Ts),
    \+(function_word(W)),
    Ws=[W],
    w2itn(Ws,GI,T,_). % word to synset and type
```

Note that we have avoided function words (auxiliary verbs included), as their semantic links to the co-occuring nouns is usually irrelevant.

8.3 Extending the Link Graph with Verb-Noun Co-occurrence Links

A second technique consists in simply adding links from verb synsets occurring in a sentence to the co-occurring noun synsets. The size of the link set is controlled by limiting the distance between the verb and noun occurrences to a small value like 3 or 4.

8.4 Extending the Link Graph with Superclass Links

We can add superclass links (hypernyms) to pull out a relevant WordNet subgraph. If we add them as outgoing links, the resulting keywords will consist in "general terms" about the document. This usually needs to be combined with filtering out dominant top-level nouns like *entity, thing, object* which are too abstract to convey interesting information about the document.

8.5 Extending the Link Graph with Subclass Links

Subclass links make sense as incoming links as they provide additional weight to synsets occuring in the text. Some filtering is required to only provide a small number per synset as this tends to bring explosive growths to the link graph, and as such it can dilute more relevant links.

8.6 Extending the Link Graph with Domain Links

Domain links have been added to WordNet 2.0 as a first step meant to complement the classification of synsets based on the "ontology" in which a given synset is relevant to. While they usually add a small number of links, their use as incoming links tends to help focusing on a dominant field which helps both disambiguation and extraction of keywords.

8.7 Customizing Abstraction Operators

Depending on the lexical category of a given meaning, different abstraction operators can be used for fine tuning the results to what a human reader would consider relevant. Our customized *generalized hypernym* abstraction operator follows the following algorithm, detailed per lexical category:

- **nouns**: hypernym links
- **verbs**: hypernym links, and hypernym links following causality and entailment links
- **adjectives**: attribute links to nouns followed by hypernym links and reverse attribute links back to adjectives
- **adjective satellites**: synonymy links to adjectives, followed by adjective abstractions as previously described

- **adverbs**: pertinence links to adjectives from which adverbs have been derived, followed by adjective abstractions and links back to adverbs

The generalized hypernym relation is used on a graph which is then extended with inferred semantic links.

The following variant of the previous algorithm can be used for extracting keyword and extracting key sentences from a text document:

1. Build the synset graph using the generalized hypernym relation.
2. For each word phrase in the text select the highest ranked synset as a disambiguation of the meaning of the phrase.
3. Compute the rank of each sentence as the sum of the ranks of its highest ranked synsets and divide the resulting rank by the number of synsets used in the computation.
4. Pick a subset of the top ranked synsets
5. Pick the highest probability WordNet word phrase that represents each top ranked synset as a *keyword*
6. Pick a subset of the top ranked sentences as the *summary*.

8.8 Heuristics for Balancing the Addition of Links

The fine-tuning of the right balance of links requires an automated search of a fairly large parameter space. However, our hand-crafted balance which simply ensures that comparable numbers of links are added through the heuristics previously described has performed intuitively well on a variety of texts.

8.9 Keyword and Sentence Extraction Experiments

While the use of PageRank as a keyword and key sentence extractor requires more work to get close to human performance, we have run some experiments on the Brown Corpus data.

The first approach attaches to each word phrase the value of its highest ranked synset.

For instance in the case of the file `br-c01` this brings out the following set:

```
[people] [air] [day] [breath] [manse]
[performance] [personality] [information]
```

The highest ranked synsets corresponding to noun phrases will provide the following set of "concept" synsets, which we chose to represent through their highest frequency word phrase associated by WordNet:

```
[person] [location] [people] [air]
[day] [breath] [thing] [mansion]
```

We have experimented with extracting high ranked sentences from a variety of Web documents and will discuss our results in a future paper.

9 Related Work

Traditional methods of scientific citation analysis have been directed to produce measures of the importance and impact indicative of authority [8] or influence [20]. However, graphs like the WWW or WordNet are not built on the same principles of peer-review and refereeing with their absence of "control" over what peers (or authorities) directly endorse other peers (or authorities). Kleinberg [11] reviews the various approaches to ranking pages in the WWW environment, discussing index nodes and reference nodes-as well as measures of centrality based on node-to-node distances. He has designed a technique for "locating high-quality information related to a broad search topic on the WWW, based on a structural analysis of the link topology surrounding authoritative pages on the topic". His HITS algorithm suggests an alternative to PageRank which selectively ranks Web pages as *authorities* (pages where high quality specialized information is found in a given field) and *hubs* (where high quality links towards other pages are found) [11].

Interestingly enough, Kleinberg's paper [11] is explained in the context of metasearch, and as such, it does not assume the availability of the complete Web graph as a local data object. A promising adaptation of the HITS algorithm for text mining would separate noun and verb vertices and examine noun to verb co-occurrences in a corpus as *links* between their corresponding synsets, with the idea of automatically deriving semantic roles similar to those in FrameNet [1].

While papers like [26] extend results about the *small-world* properties of the World Wide Web graph to semantic networks like WordNet, at our best knowledge, corroborated by extensive CiteSeer search, this paper is likely to be the first attempt to evaluate the use of PageRank-style link-analysis algorithms on semantic networks for automatic analysis of text documents.

Acknowledgement

We thank the referees of the CICLOPS'04 workshop for their suggestions and comments.

10 Conclusion

We have shown that PageRank-style algorithms, originally designed for content-independent Web link analysis to WordNet-based synset graphs are usable for disambiguation and content extraction tasks from natural language texts. The paper has described a new approach to knowledge-based word-sense disambiguation that relies on PageRank-style algorithms applied on semantic networks, and we have shown that the accuracy achieved through our exceeds the performance obtained by other dictionary-based algorithms. Future work will focus on a mechanism allowing to automatically try out various combinations of inferred WordNet relations on a larger corpus of annotated data and various applications to analysis of text-documents and in particular, their use as a filtering mechanism for improving metasearch algorithms on Web documents.

References

[1] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet project. In C. Boitet and P. Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 86–90, San Francisco, California, 1998. Morgan Kaufmann Publishers.

[2] P. Boldi and S. Vigna. The webgraph framework i: Compression techniques, 2003. http://citeseer.nj.nec.com/boldi03webgraph.html.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998. http://citeseer.nj.nec.com/brin98anatomy.html.

[4] C. Felbaum. *Wordnet, an Electronic Lexical Database for English.* Cambridge: MIT Press, 1998.

[5] E. Figa and P. Tarau. Lexical Inference Mechanisms for Text Understanding and Classification. In R. J. Todd, editor, *Proceedings of the 66th Annual Meeting of American Society for Information Science and Technology*, pages 165–173, Long Beach, California, Oct. 2003.

[6] E. Figa and P. Tarau. Story Traces and Projections: Exploring the Patterns of Storytelling. In N. Braun and U. Spierling, editors, *TIDSE'2003*, Darmstadt, Germany, Mar. 2003.

[7] W. Gale, K. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Harriman, New York, 1992.

[8] E. Garfield. Citation analysis as a tool in journal evaluation. Science 178, 1972.

[9] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Extrapolation methods for accelerating pagerank computations, 2003. http://citeseer.nj.nec.com/kamvar03extrapolation.html.

[10] A. Kiryakov, K. Simov, and M. Dimitrov. OntoMap: ontologies for lexical semantics. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *Proceedings of the Euroconference Recent Advances in Natural Language Processing (RANLP-2001)*, pages 142–148, Tzigov, Bulgaria, 2001. http://citeseer.nj.nec.com/kiryakov01ontomap.html.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999. http://citeseer.nj.nec.com/kleinberg99authoritative.html.

[12] M. Lesk. 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 1986.

[13] R. Mihalcea and D. Moldovan. An iterative approach to word sense disambiguation. In *Proceedings of FLAIRS-2000*, pages 219–223, Orlando, FL, May 2000.

[14] R. Mihalcea and D. Moldovan. eXtended Wordnet: progress report, 2001.

[15] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University, July 1990.

[16] G. Miller, C. Leacock, T. Randee, and R. Bunker. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, New Jersey, 1993.

[17] H. Ng and H. Lee. Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, Santa Cruz, 1996.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[19] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics CICLING-2003*, Mexico City, February 2003.

[20] G. Pinski and F. Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. Inf. Proc. Manage. 12, pages 297-312.

[21] R. Rada, H. Mili, E. Bickell, and B. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19:17–30, Jan/Feb 1989.

[22] S. Raghavan and H. Garcia-Molina. Representing web graphs, 2003. http://citeseer.nj.nec.com/article/raghavan03representing.html.

[23] P. Resnik. Selectional preference and sense disambiguation. In *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?*, Washington DC, April 1997.

[24] J. F. Sowa. Relating templates to logic and language. In M. T. Pazienza, editor, *Information Extraction: Towards Scalable, Adaptable Systems, LNAI 1714*, pages 76–94. Springer-Verlag, 1999. http://users.bestweb.net/ sowa/direct/template.htm.

[25] J. Stetina, S. Kurohashi, and M. Nagao. 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 1998.

[26] M. Steyvers and J. Tenenbaum. Small worlds in semantic networks. http://citeseer.nj.nec.com/438759.html.

[27] P. Tarau. The Jinni 2004 Prolog Compiler: a High Performance Java and .NET based Prolog for Object and Agent Oriented Internet Programming. Technical report, BinNet Corp., 2004. URL: http://www.binnetcorp.com/download/jinnidemo/JinniUserGuide.html.

[28] S. Tyagi and P. Tarau. A Most Specific Method Finding Algorithm for Reflection Based Dynamic Prolog-to-Java Interfaces. In I. Ramakrishan and G. Gupta, editors, *Proceedings of PADL'2001*, Las Vegas, Mar. 2001. Springer-Verlag.

[29] D. Yarowsky. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop*, 1993.

[30] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, pages 189–196, Cambridge, MA, 1995 1995.