Semantic Document Engineering with WordNet and PageRank

Paul Tarau [*]
Department of Computer
Science and Engineering
University of North Texas
P.O. Box 311366,Denton,
Texas 76203
tarau@cs.unt.edu

Rada Mihalcea [†]
Department of Computer
Science and Engineering
University of North Texas
P.O. Box 311366,Denton,
Texas 76203
rada@cs.unt.edu

Elizabeth Figa [‡]
School of Library and
Information Sciences
University of North Texas
P.O. Box 311366,Denton,
Texas 76203
efiga@lis.admin.unt.edu

ABSTRACT

This paper describes Natural Language Processing techniques for document engineering 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 on standard benchmarks.

Keywords

Word Sense Disambiguation, PageRank-style graph algorithms, WordNet, semantics-based document processing, Logic Programming, Natural Language Processing

1. INTRODUCTION

Google's PageRank [1, 10] link-analysis algorithm and variants like Kleinberg's HITS algorithm [6] 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 [8, 2] 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 [8] as a "machine readable lexical database organized by meanings". WordNet maps word forms and word meanings as a many-to-many relation, which indicates that some word forms can have several different meanings, and some meanings can be expressed by different word 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).

The WordNet database [8] 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 synsets in which they occur provides fast access from lexical forms 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 effi-

[*] Acknowledges support from UNT and BinNet Corporation, `http://www.binnetcorp.com`.

[†] Supported by NSF and UNT research grants.

[‡] Supported by UNT research grant.

SAC'05 March 13-17, 2005, Santa Fe, New Mexico, USA
Copyright 2005 ACM 1-58113-964-0/05/0003 ...$5.00.

ciently, if needed, at runtime. We also collect frequency information and word forms not present in the form of WordNet entries.

Building the WordNet Graph. We have converted the refactored Prolog WordNet database described in [3] to a Java graph representation using Jinni 2004's [11] built-in graph processing libraries.

3. IMPLEMENTING THE PAGERANK ALGORITHM

We will now shortly describe the PageRank algorithm, following [10, 1].

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:

$$PR(P_0) = (1 - d) + d * \sum_{i=1..N} \frac{PR(P_i)}{|OUT(P_j)|}$$

Note that somewhat faster or more compact (but significantly more complex) implementations have been recently suggested [4], 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 [10] which is now part of Jinni 2004's [11] graph processing API. Starting from arbitrary values assigned to each node, the code iterates the PR score computation until convergence below a given threshold is noticed. It has been proven in [10] 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.

4. A VIEW FROM THE TOP: PAGE-RANKING 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* [5] 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]
```

This points out that the dominant verb ontology implicit in WordNet is oriented toward describing change, movement and interaction. Verbs of creation are also highly ranked, for instance the synset for `[make] [create]` is one of the top synsets, with a rank score of 106.

5. SPECIALIZING PAGERANK FOR 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 (two concepts that share the same hypernym). It is also worth mentioning the composite sense discriminating 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.

We can build the subgraph by selecting only a given relation, and by ranking the 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.

6. 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.

6.1 Related Work

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 [12]), 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.

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.

6.2 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 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 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.

6.3 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.

In addition to the link graph, two additional graphs are built, which are used to identify important keywords and sentences in the text (see Section 7.2).

- 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.

6.4 Experimental Evaluation

To evaluate the performance of the word sense disambiguation algorithm, we use a subset of SemCor [9] – a relatively 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 [7], 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.

7. 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.

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 PageRank graph, as illustrated in the following example from the Brown corpus, 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]])
```

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.

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).

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.

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.

Extending the Link Graph with Subclass Links. Subclass links make sense as incoming links as they provide additional weight to synsets occurring 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.

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.

7.1 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.

7.2 Keyword and Sentence Extraction Experiments

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.*

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]
```

Results obtained during preliminary experiments suggests the validity of this graph-based approach for keyword and sentence extraction. We are currently working on validating these initial findings through more extensive evaluations on standard data sets.

8. CONCLUSION

We have shown that PageRank-style algorithms, originally designed for content-independent Web link analysis, can be applied to WordNet-based synset graphs, resulting in efficient algorithms for concept reranking, word sense disambiguation, and content extraction from natural language documents. We have described and evaluated a new approach for knowledge-based word-sense disambiguation relying on PageRank-style algorithms applied on semantic networks, and showed that the accuracy achieved by our algorithms exceeds the one of previously proposed dictionary-based methods. 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.

9. REFERENCES

[1] 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.

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

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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. http://www.binnetcorp.com/download/jinnidemo/JinniUserGuide.html.

[12] 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.