“PageRank on Semantic Networks, with Application to Word Sense Disambiguation”

A paper by Rada Mihalcea, Paul Tarau, Elizabeth Figa

Amin Kiem
amin.kiem@gmail.com

HS Graph-Based Methods for NLP (Dr. Simone Paolo Ponzetto)
Ruprecht-Karls-Universität Heidelberg
Seminar für Computerlinguistik

2010-12-09

What this paper is about

The PageRank[a] algorithm is applied to a graph built of meaningfully interconnected WordNet synsets to perform *word sense disambiguation (WSD)*. → **"pageranking" WordNet**

[a] (Brin and Page, 1998)

Outline

1. Word Sense Disambiguation
2. Preliminaries: PageRank and WordNet
3. The PageRank Disambiguation Algorithm
4. Related Algorithms and Improvements
5. Evaluation
6. Conclusion

1. Word Sense Disambiguation
    - Motivation
    - Supervised vs. Unsupervised Learning
    - Open-text Knowledge-based Algorithms
2. Preliminaries: PageRank and WordNet
3. The PageRank Disambiguation Algorithm
4. Related Algorithms and Improvements
5. Evaluation
6. Conclusion

WSD: Motivation

Task

assigning the most appropriate meaning to a polysemous word within a given context

Application:

- machine translation
- knowledge acquisition
- common sense reasoning
- . . .

WSD: supervised vs. unsupervised learning

Supervised learning

labeled data (e.g. words with sense tags) gets transformed into feature vectors which are then used in an automatic learning process

- pro: higher performance
- con: only usable for words for which sense tagged data can be provided by large corporas

Unsupervised Learning

works with unlabeled data and therefore on any unseen text; aka the *open-text knowledge-based approach*

- pro: larger coverage; applicable to *all words* in open-text
- con: lower performance

WSD: open-text knowledge-based algorithms

Four Main Types

1. **Lesk algorithms**
2. Semantic similarity
3. Selectional preferences
4. Heuristic-based methods

1. Word Sense Disambiguation

2. Preliminaries: PageRank and WordNet
    - The PageRank Algorithm
    - WordNet
    - Semantic Relations

3. The PageRank Disambiguation Algorithm

4. Related Algorithms and Improvements

5. Evaluation

6. Conclusion

Preliminaries: The PageRank Algorithm

$=$ iterative graph-based ranking algorithm

- Let $G = (V, E)$ be a graph with *V*: a set of vertices and *E*: a set of edges from *V x V*
- For a given vertex V_i let ...
  - $In(V_i)$ be the set of vertices that point to V_i
  - $Out(V_i)$ be the set of outgoing edges from V_i
- The PageRank of V_i is given by

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

- *d*: dumping factor set between 0 and 1 (the authors used 0.85)

PageRank decides how important a vertex is within a graph by giving each vertex a score. 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.

Preliminaries: WordNet as a Graph (I)

About WordNet[1]

- lexical database for English
- nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (**synsets**)
  - each synset expresses a distinct concept
  - synsets are interlinked by means of conceptual-semantic and lexical relations (like *IS-A*, *HAS-A*, *PART-OF* . . .)

What to tell your grandmother

WordNet is a network of meaningfully related words and concepts.

[1] http://wordnet.princeton.edu/

Preliminaries: WordNet as a Graph (II)

Graph representation

- vertices = **synsets**
- edges = **semantic relations (SR)** or sets of semantic relations

The graph can be constructed directed as well as undirected.
Every graph representation of an input text is a subgraph of the WordNet graph. (we'll see *Text Synset Graphs* in a minute)

Preliminaries: Semantic Relations (= *edges*)

Basic semantic relations

→ Explicitly encoded by WordNet:
hypernymy, hyponymy, meronymy, holonymy, entailment, causality, attribute, pertainimy

Derived semantic relations

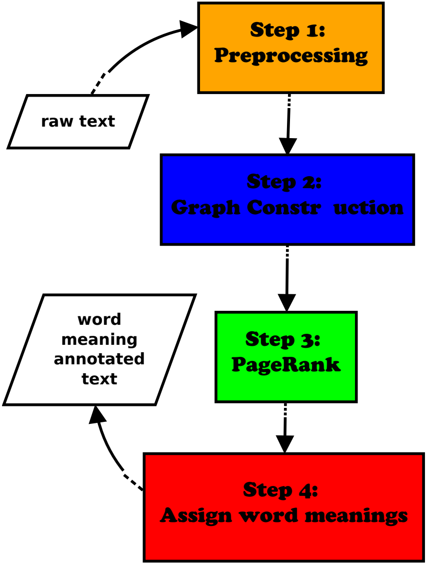
→ Combinations of two or more basic semantic relations

Example:

- *hypernymy* + *hyponymy* = coordinate-relation (identifies synsets that share the same hypernym: *dog#1* and *wolf#1* share *canine#1*)
- xlink-relation: combines all basic SR's and the coordinate-relation to one relation

1. Word Sense Disambiguation
2. Preliminaries: PageRank and WordNet
3. The PageRank Disambiguation Algorithm
4. Related Algorithms and Improvements
5. Evaluation
6. Conclusion

The PageRank Disambiguation Algorithm



raw text

Step 1: Preprocessing

Step 2: Graph Construction

Step 3: PageRank

Step 4: Assign word meanings

word meaning annotated text

Step 1: Preprocessing

Task: *Suit-up!* the input data

1. text gets tokenized and annotated with PoS-Tags
2. identification of collocations
3. identification of named entities

Step 2: Graph construction

Task: Construct a graph of the input text (*Text Synset Graph*)

1. **add vertices**
   for all open-class words in the input text, add all identified WordNet synsets as vertices in the graph (named entities and modal/auxiliary verbs are not considered)
2. **add edges**
   add an edge between all vertices in the graph that can be linked by a given semantic relation

The *Text Synset Graph* is a subgraph of WordNet.

Step 3: PageRank

Task: Score the vertices

1. **initialize graph**
   a small value is assigned to each vertex (Rada et al. used an initial value of 1)
2. **iterate until convergence**
   takes about 25-30 iterations

Step 4: Assign word meanings

Task: dito.

1. **Win: a word-sense is uniquely identifiable**
   after iteration over the ambiguous words in the text the synset that has the highest PageRank score was found
2. **Fail: no word-sense can be found**
   assign the word with a random sense (*uninformed algortihm*) or with the first sense in WordNet (*informed algortihm*)

1. Word Sense Disambiguation
2. Preliminaries: PageRank and WordNet
3. The PageRank Disambiguation Algorithm
4. Related Algorithms and Improvements
5. Evaluation
6. Conclusion

Related Algorithms: The Lesk Algorithm (I)

Classic LESK

- first introduced by Michael E. Lesk, 1986
- one of the first algorithms capable of disambiguating *all words* in open text
- uses a dictionary for disambiguation (dictionary entries = senses)

Procedure

1. compare the different senses for the current word with the senses of all other words in context
2. choose the sense with the highest overlap

Related Algorithms: The Lesk Algorithm (II)

Simplified LESK

→ Problem *Classic LESK*: Explosion of word sense combinations

Example

"I saw a man who is 108 years old and can still walk and tell jokes".

- count of senses per word: see(26), man(11), year(4), old(8), can(5), still(4), walk(10), tell(8), joke(3)
- possible sense combinations: **43.929.600**

Simplification

Directly compare the dictionary entry of one word with its context words without looking at every context words dictionary entry.

Related Algorithms: Most Frequent Sense

→ simple algorithm: just picks the most frequent sense for any given word as the correct one

- possible because WordNet keeps track of the frequency of each word meaning within a sense-annotated corpus

Improvements

(I) Combining PageRank and Lesk

1. use LESK to provide a basic ordering of senses for a given word
2. let the PageRank algorithm run; if two senses have similar PageRank values, they will keep their LESK ordering, otherwise PageRank overrides LESK

(II) Combining PageRank with Most Frequent Sense

To disambiguate a word. . .

1. order senses according to their WordNet frequency
2. calculate a combined rank for a sense from its PageRank and its WordNet frequency by multiplication[2]

[2] or with a special formula, see (Mihalcea et al., 2004), section 5.4 for details

1. Word Sense Disambiguation

2. Preliminaries: PageRank and WordNet

3. The PageRank Disambiguation Algorithm

4. Related Algorithms and Improvements

5. Evaluation
    - The Data
    - Uninformed vs. Informed Algorithms
    - Results
    - Discussion

6. Conclusion

Evaluation: The Data

Algorithms are tested on a benchmark of sense-annotated texts (manually annotated by a lexicographer) taken from the following datasets:

- SemCor: five randomly selected files from different topics
- SENSEVAL-2 dataset for the *English all words task*

Average size of a file is 600-800 open-class words.

Uninformed vs. Informed Algorithms

Uninformed Algorithms

The **sense ordering** provided by the dictionary (WordNet) is not used by the algorithms. To ensure this point, the senses get shuffled with a uniform distribution.

Random baseline: assigns a random sense to each word in the text

Informed Algorithms

The algorithms use the **sense ordering** provided by WordNet.

"Most frequent sense" as baseline: assigns the most frequent sense to each word in the text

Evaluation Results (I): Uninformed Algorithms

Table 1: WSD accuracy **without** integration of WordNet sense order

	Size(words)	Random	Lesk	PageRank	PageRank+Lesk
SEMCOR					
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%
SENSEVAL-2					
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%

Evaluation Results (II): Informed Algorithms

Table 2: WSD accuracy **with** integration of WordNet sense order

	Size(words)	MFS	Lesk	PageRank	PageRank+Lesk
SemCor					
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%
Senseval-2					
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%

Discussion

Some observations

- The new algorithm always exceeds the baseline by a large margin
- No matter if it uses word sense ordering or not, it always performs better than LESK
- The combined method (PageRank + LESK) always outperforms all other algorithms (perhaps because of an increase of redundancy through integration of an extra knowledge element)

1. Word Sense Disambiguation
2. Preliminaries: PageRank and WordNet
3. The PageRank Disambiguation Algorithm
4. Related Algorithms and Improvements
5. Evaluation
6. Conclusion

Conclusion

- The authors proposed a new PageRank-based, unsupervised knowledge-based WSD algorithm that **outperforms** all other known knowledge-based algorithms.
- The uninformed version of the algorithm is **highly portable to other languages**, because it only relies on dictionary information (no sense frequencies needed).

Literature

- R. Mihalcea, P. Tarau and E. Figa (2004). *PageRank on Semantic Networks, with application to Word Sense Disambiguation.* In Proceedings of COLING '04.
- M. Lesk (1986). *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone.* In SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, pages 24-26, New York, NY, USA. ACM.
- S. Brin and L. Page (1998). *The anatomy of a large-scale hypertextual Web search engine.* Computer Networks and ISDN Systems, 30(1–7):107–117.