Networks and Natural Language Processing

Dragomir Radev
University of Michigan
radev@umich.edu

Rada Mihalcea
University of North Texas
rada@cs.unt.edu

Abstract

Over the last few years, a number of areas of natural language processing have begun applying graph-based techniques. These include, among others, text summarization, syntactic parsing, word sense disambiguation, ontology construction, sentiment and subjectivity analysis, text clustering. In this paper, we present some of the most successful graph-based representations and algorithms used in language processing, and try to explain how and why they work.

1 Introduction

In a cohesive text, language units – whether they are words, phrases, or entire sentences – are connected through a variety of relations, which contribute to the overall meaning of the text and maintain the cohesive structure of the text and the discourse unity. Since the early ages of artificial intelligence, associative or semantic networks have been proposed as representations that enable the storage of such language units and the relations that interconnect them, and which allow for a variety of inference and reasoning processes, simulating some of the functionalities of the human mind. The symbolic structures that emerge from these representations correspond naturally to graphs – where text constituents are represented as vertices, and their interconnecting relations form the edges in the graph.

The last decade has brought on a number of exciting research papers that apply graph-based methods to an increasingly large range of natural language problems, ranging from lexical acquisition to sentence parsing to word sense disambiguation and text summarization. In this paper, we overview several of these methods and their application to natural language processing. To reflect the fact that the algorithms and representations originate in different communities – natural language processing and graph-theory – we will be using a dual vocabulary to describe these methods: networks are graphs, nodes are vertices, and links are edges.

In terms of **graph-based representations**, depending on the natural language processing application, a variety of node and edge types have been used. Text units of various sizes and characteristics can be added as vertices in the graph, e.g. words, collocations, word-senses, entire sentences, or even entire documents. Note that the graph-nodes do not have to belong to the same category. For example, both sentences and words can be added as vertices in the same graph. Edges can represent cooccurrence (e.g., two words that appear in the same sentence or in the same dictionary definition), collocation (e.g., two words that appear immediately next to each other or which may be separated by a conjunction), syntactic structure (e.g., the parent and child in a syntactic dependency), lexical similarity (e.g., cosine between the vector representations of two sentences).

In terms of **graph-based algorithms**, the main methods used so far can be classified into: (1) semi-supervised classification (Zhu and Ghahramani, 2002; Zhu and Lafferty, 2005; Toutanova et al., 2004; Radev, 2004; Otterbacher et al., 2005), where random walks or relaxation are applied on mixed sets of labeled and unlabeled nodes;

(2) network analysis (Masucci and Rodgers, 2006; Caldeira et al., 2006), where network properties such as diameter, centrality, etc., are calculated; (3) graph-based clustering methods (Pang and Lee, 2004; Widdows and Dorow, 2002), such as min-cut methods; (4) minimum spanning-tree algorithms (McDonald et al., 2005).

In this paper, we overview several graph-based approaches for natural language processing tasks, which we broadly group into three main categories. First, we review research work done in the area of syntax, including syntactic parsing, prepositional attachment and co-reference resolution. We then describe methods used in lexical semantics, including word sense disambiguation, lexical acquisition, and sentiment and subjectivity analysis. Finally, we review several natural language processing applications that rely on graph methods, including text summarization, passage retrieval and keyword extraction.

2 Syntax

In this section we will discuss three papers, addressing methods for syntactic parsing (McDonald et al., 2005), prepositional attachment (Toutanova et al., 2004), and coreference resolution (Nicolae and Nicolae, 2006).

2.1 Dependency Parsing

McDonald et al. (McDonald et al., 2005) take an unconventional approach to sentence parsing. They start by realizing that each dependency tree of a sentence is a directed subgraph of the full graph linking all words in the sentence. An approach like theirs would not work on the more widely known constituent trees as they contain non-terminals. In dependency parsing, each sentence is represented as a tree, the root of which is typically the main predicate of the sentence (or it is a dummy node labeled root of which the main predicate is the sole child) and in which edges are used to connect each word to its dependency parent. For example, in the sentence *John likes green apples*, the main predicate is *likes*, which takes two arguments: the liker (*John*) and the liked (*apples*). Finally, since *green* modifies *apples*, it is added to the tree as a child of *apples*. The final tree looks like this:

```
[likes [John] [apples [green]]].
```

McDonald et al. build a full graph from the input sentence and then associate a score with each potential directed subtree of that graph which is equal to the sum of the scores of all edges that it includes. The score for each edge in the original graph is the product of a weight vector w and a feature representation of the edge $f(i,j)$. The goal of their parser is to find the tree with the highest score. Note that punctuation, including the final period of a sentence is used in the parsing process.

The Chu-Liu-Edmonds (CLE) algorithm (J. and Liu, 1965; Edmonds, 1967) is used to find maximum spanning trees (MST) in directed graphs. The algorithm involves the following procedure: each node picks the neighbor with the highest score. The result is either a spanning tree or it has cycles. The CLE method collapses each such cycle into a single node and recomputes the scores of each edge incident on such a cycle. It can be shown that the MST built on the collapsed graph is the same as the MST computed on the original graph; the algorithm can be run in $O(n^2)$.

Lets consider an example consistent with McDonald et al. The sentence that needs to be parsed is *John likes green apples*. The corresponding graph is shown in the leftmost graph in Figure 1, and each node in the graph corresponds to a word in the sentence. After the first iteration of the algorithm, no tree is found that covers all nodes, so the two closest nodes are collapsed, resulting in the second graph in Figure 1. The process continues until the entire graph gets reduced to a single node through a series of iterations.

After all nodes have been collapsed into one, the MST is constructed by reversing the procedure and expanding all nodes into their constituents. The end result for this example is shown in Figure 2.

McDonald et al. achieve state of the art results with their parser on a standard English data set and better than state of the art results on Czech (a free word order language).

2.2 Prepositional Phrase Attachment

Prepositional phrase (PP) attachment is one of the most challenging problems in parsing. English grammar allows a prepositional phrase such as *with*

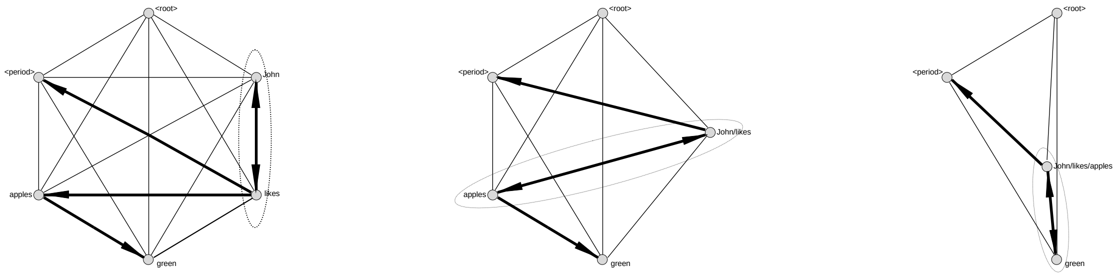

Figure 1: The graphs produced by intermediate steps of the MST algorithm.

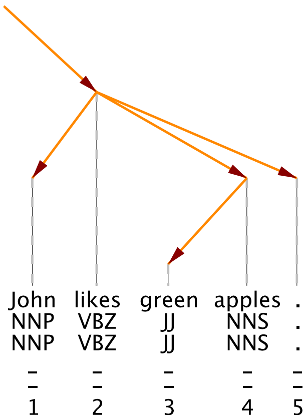

Figure 2: Output of the MST parser.

to attach either to the main predicate of a sentence or to the noun phrase immediately preceding it. For example, *I ate pizza with olives* is an example of low (nominal) attachment whereas *I ate pizza with a knife* is an example of high (verbal) attachment. In naturally occurring English text, both types of attachment are commonly seen.

Toutanova et al. (Toutanova et al., 2004) address the problem of PP attachment by casting it as a semi-supervised learning process on graphs. Each node of the graph corresponds to a verb or a noun. Two nodes are connected if they appear in the same context (e.g., the verb nodes *hang* and *fasten* are connected because they both appear in the expressions *with a nail*. In a similar way, the noun nodes *nail* and *rivet* are connected to each other. Many other types of connections (more than 10 types, including links between words with the same root form, synonyms, etc.) are described in the paper. The algorithm then proceeds with a random walk on the graph until convergence. The evaluation is performed on a standard test set from the Penn Treebank. The results reported in the paper show a performance of 87.54% classification accuracy, which is very close to the upper bound corresponding to human performance (88.20%).

2.3 Coreference Resolution

Coreference resolution is defined as the problem of identifying relations between entity references in a text, whether they are represented by nouns or pronouns. Typical algorithms for coreference resolution attempt to identify chains of references by using rule-based systems or machine learning classifiers. In recent work, (Nicolae and Nicolae, 2006) introduced a graph-based approach to coreference resolution, which attempts to approximate the correct assignment of references to entities in a text by using a graph-cut algorithm.

A separate graph is created for each NIST-defined entity type, including *Person*, *Organization*, *Location*, *Facility* and *GPE*. Next, weighted edges are drawn between the entity references, where the weights correspond to the confidence of a coreference relation. Finally, a partitioning method based on min-cut is applied on these graphs, which will separate the references corresponding to the same entity. When evaluated on standard benchmarks for coreference resolution, the graph-based algorithm was found to lead to state-of-the-art performance, improving considerably over previous algorithms.

3 Lexical Semantics

There has been growing interest in the automatic semantic analysis of text, to support natural language processing applications ranging from machine translation and information retrieval, to question answering and knowledge acquisition. A significant amount of research has been carried out in this area, including work on word sense disambiguation, semantic role labeling, textual entailment, lexical acquisition, and semantic relations. In this section, we will review several methods based on graph representations and algorithms that have been used to address various tasks in automatic semantic analysis.

3.1 Lexical Networks

One of the largest graph representations constructed to support a natural language processing task is perhaps the graph model proposed by Widdows and Dorow for unsupervised lexical acquisition (Widdows and Dorow, 2002). The goal of their work is to build semantic classes, by automatically extracting from raw corpora all the elements belonging to a certain semantic category such as *fruits* or *musical instruments*.

The method starts by constructing a large graph consisting of all the nouns in a large corpus (British National Corpus, in their case), linked by the conjunction *and* or *or*. A cutoff value is used to filter out rare words, resulting in a graph of almost 100,000 nouns, linked by more than half-million edges. To identify the elements of a semantic class, first a few representative nouns are manually selected and used to form a seed set. Next, in an iterative process, the node found to have the largest number of links with the seed set in the co-occurrence graph is selected as potentially correct, and thus added to the seed set. The process is repeated until no new elements can be reliably added to the seed set. Figure 3 shows a sample of a graph built to extract semantic classes.

An evaluation against ten semantic classes from WordNet indicated an accuracy of 82% which, according to the authors, was an order of magnitude better than previous work in semantic class extraction. The drawback of their method is the low coverage, given that the method is limited to those words found in a conjunction relation. However, whenever applicable, the graph representation has the ability to precisely identify the words belonging to a semantic class.

Another research area related to the work of (Widdows and Dorow, 2002) is the study of lexical network properties carried out by (Ferrer-i-Cancho and Sole, 2001). By building very large lexical networks of nearly half-million nodes, with more than ten million edges, constructed by linking words appearing in English sentences within a distance of at most two words, they proved that complex system properties hold on such co-occurrence networks.

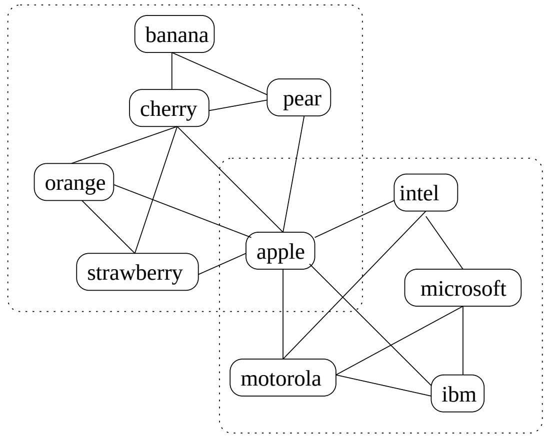

Figure 3: Lexical network constructed for the extraction of semantic classes.

Specifically, they observed a small-world effect, with a relatively small number of 2-3 jumps required to connect any two words in the lexical network. Additionally, it has also been observed that the distribution of node degrees inside the network is scale-free, which reflects the tendency of a link to be formed with an already highly connected word. Perhaps not surprisingly, the small-world and scale-free properties observed over lexical networks automatically acquired from corpora were also observed on manually-constructed semantic networks such as WordNet (Sigman and Cecchi, 2002; Steyvers and Tenenbaum, 2005).

3.2 Semantic Similarity and Relatedness

Graph-based algorithms have also been successfully used in identifying word similarity and relatedness. A large class of methods for semantic similarity consists of metrics calculated on existing semantic networks such as WordNet and Roget, by applying, for instance, shortest path algorithms that identify the closest semantic relation between two input concepts (Leacock et al., 1998).

More recently, an algorithm based on random walks was proposed by Hughes and Ramage (Hughes and Ramage, 2007). Briefly, in their method, the PageRank algorithm is used to calculate the stationary distribution of the nodes in the WordNet graph, biased on each of the input words in a given word pair. Next, the divergence between these distributions is calculated, which reflects the relatedness of the two words. When evaluated on standard word relatedness data sets, the method was found to improve significantly over previously proposed algorithms for semantic relatedness. In fact, their best performing measure came close to the upper bound represented by the inter-annotator agreement on these data sets.

3.3 Word Sense Disambiguation

Another topic of interest in lexical semantics is word sense disambiguation, defined as the problem of identifying the most appropriate meaning of a word given its context. Most of the work in this area assumes the availability of a predefined sense inventory such as WordNet, and consists of methods that can be broadly classified as knowledge-based, supervised or semi-supervised.

A graph-based method that has been successfully used for semi-supervised word sense disambiguation is the label propagation algorithm (Niu et al., 2005). In their work, Niu and colleagues start by constructing a graph consisting of all the labeled and unlabeled examples provided for a given ambiguous word. The word sense examples are used as nodes in the graph, and weighted edges are drawn by using a pairwise metric of similarity. On this graph, all the known labeled examples (the seed set) are assigned with their correct labels, which are then propagated throughout the graph across the weighted links. In this way, all the nodes are assigned with a set of labels, each with a certain probability. The algorithm is repeated through convergence, with the known labeled examples being reassigned with their correct label at each iteration. In an evaluation carried out on a standard word sense disambiguation data set, the performance of the algorithm was found to exceed the one obtained with monolingual or bilingual bootstrapping. The algorithm was also found to perform better than SVM when only a few labeled examples were available.

Graph-based methods have also been used for knowledge-based word sense disambiguation. In (Mihalcea et al., 2004; Sinha and Mihalcea, 2007), Mihalcea et al. proposed a method based on graphs constructed based on WordNet. Given an input text, a graph is built by adding all the possible senses for the words in the text, which are then connected on the basis of the semantic relations available in the WordNet lexicon (e.g., synonymy, antonymy, etc.). For instance, Figure 4 shows an example of a graph constructed over a short sentence of four words.

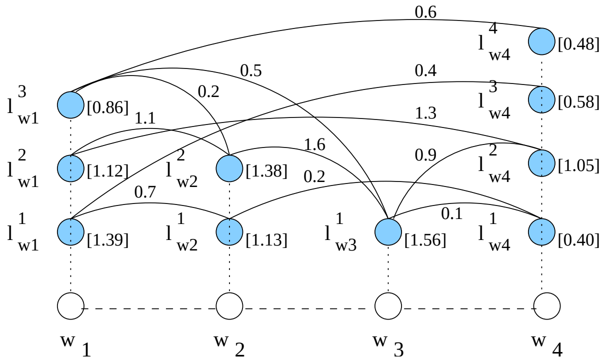

Figure 4: Graph constructed over the word senses in a sentence, to support automatic word sense disambiguation.

A random-walk applied on this graph results in a set of scores that reflects the “importance” of each word sense in the given text. The word senses with the highest score are consequently selected as potentially correct. An evaluation on sense-annotated data showed that this graph-based algorithm was superior to alternative knowledge-based methods that did not make use of such rich representations of word sense relationships.

In follow-up work, Mihalcea developed a more general graph-based method that did not require the availability of semantic relations such as those defined in WordNet. Instead, she used derived weighted edges determined by using a measure of lexical similarity among word sense definitions (Mihalcea, 2005), which brought generality, as the method is not restricted to semantic networks such as WordNet but it can be used on any electronic dictionaries.

Along similar lines with (Mihalcea et al., 2004), Navigli and Lapata carried out a comparative evaluation of several graph connectivity algorithms applied on word sense graphs derived from WordNet

(Navigli and Lapata, 2007). They found that the best word sense disambiguation accuracy is achieved by using a closeness measure, which was found superior to other graph centrality algorithms such as indegree, PageRank, and betweenness.

3.4 Sentiment and Subjectivity

Sentiment and subjectivity analysis is an area related to both semantics and pragmatics, which has received a lot of attention from the research community. A method based on graphs has been proposed by Pang and Lee (Pang and Lee, 2004), where they show that a min-cut graph-based algorithm can be effectively applied to build subjective extracts of movie reviews.

First, they construct a graph by adding all the sentences in a review as nodes, and by drawing edges based on sentence proximity. Each node in the graph is initially assigned with a score indicating the probability of the corresponding sentence being subjective or objective, based on an estimate provided by a supervised subjectivity classifier. A min-cut algorithm is then applied on the graph and used to separate the subjective sentences from the objective ones. Figure 5 illustrates the graph constructed over the sentences in a text, on which the min-cut algorithm is applied to identify and extract the subjective sentences.

The precision of this graph-based subjectivity classifier was found to be better than the labeling obtained with the initial supervised classifier. Moreover, a polarity classifier relying on the min-cut subjective extracts was found to be more accurate than one applied on entire reviews.

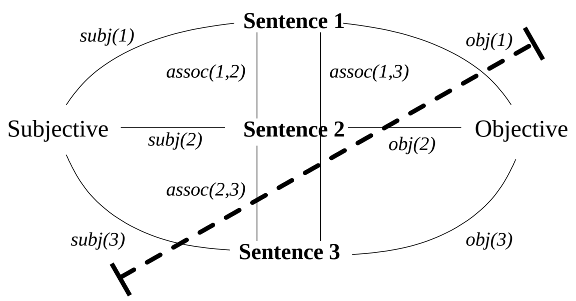

Figure 5: Subjectivity classification using a min-cut algorithm. The dotted line represents the split between subjective and objective sentences, as obtained with the min-cut algorithm.

Recent research on sentiment and subjectivity analysis has also considered the relation between word senses and subjectivity (Wiebe and Mihalcea, 2006). In work targeting the assignment of subjectivity and polarity labels to WordNet senses, Esuli and Sebastiani applied a biased PageRank algorithm on the entire WordNet graph (Esuli and Sebastiani, 2007). Similar to some extent to the label propagation method, their random-walk algorithm was seeded with nodes labeled for subjectivity and polarity. When compared to a simpler classification method, their random-walk was found to result in more accurate annotations of subjectivity and polarity of word senses.

4 Other Applications

A number of other natural language processing applications such as text summarization, passage retrieval and keyword extraction are amenable to graph-based techniques.

4.1 Summarization

One of the first graph-based methods for summarization was introduced by Allan et al (Salton et al., 1994; Salton et al., 1997). In it, they represented articles from the Funk and Wagnalls encyclopedia as graphs in which each node corresponds to a paragraph and lexically similar paragraphs are linked. A summary is then produced by starting at the first paragraph of a document and following paths defined by different algorithms that cover as much of the contents of the graph as possible.

(Erkan and Radev, 2004; Mihalcea and Tarau, 2004) take the idea of graph-based summarization further by introducing the concept of *lexical centrality*. Lexical centrality is a measure of importance (centrality) of nodes in a graph formed by linking lexically related sentences or documents. A random walk is then executed on the graph and the nodes that are visited the most frequently are selected as the summary of the input graph (which, in most cases, consists of information from multiple documents). One should note however, that in order to avoid nodes with duplicate or near duplicate content, the final decision about including a node in the summary also depends on its maximal marginal relevance as defined in (Carbonell and Goldstein, 1998). (Erkan and Radev, 2004) also builds on top of earlier summarization technology, namely the first web-accessible news summarization system, NewsInEssence (Radev et al., 2001).

An example from (Erkan and Radev, 2004) is shown in Figure 6. The input consists of 11 sentences from several news stories on related topics. Figure 7 shows the cosine similarities of all pairs of sentences while Figure 8 shows the distribution of cosines.

SNo	ID	Text
1	d1s1	Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
2	d2s1	Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.
3	d2s2	Ramadan told reporters in Baghdad that "Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it.
4	d2s3	Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq's weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.
5	d3s1	The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.
6	d3s2	Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, "will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region."
7	d3s3	Nevertheless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).
8	d4s1	The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.
9	d5s1	British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq "did not end" and that Britain is still "ready, prepared, and able to strike Iraq."
10	d5s2	In a gathering with the press held at the Prime Minister's office, Blair contended that the crisis with Iraq "will not end until Iraq has absolutely and unconditionally respected its commitments" towards the United Nations.
11	d5s3	A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

Figure 6: A cluster of 11 related sentences.

	1	2	3	4	5	6	7	8	9	10	11
1	1.00	0.45	0.02	0.17	0.03	0.22	0.03	0.28	0.06	0.06	0.00
2	0.45	1.00	0.16	0.27	0.03	0.19	0.03	0.21	0.03	0.15	0.00
3	0.02	0.16	1.00	0.03	0.00	0.01	0.03	0.04	0.00	0.01	0.00
4	0.17	0.27	0.03	1.00	0.01	0.16	0.28	0.17	0.00	0.09	0.01
5	0.03	0.03	0.00	0.01	1.00	0.29	0.05	0.15	0.20	0.04	0.18
6	0.22	0.19	0.01	0.16	0.29	1.00	0.05	0.29	0.04	0.20	0.03
7	0.03	0.03	0.03	0.28	0.05	0.05	1.00	0.06	0.00	0.00	0.01
8	0.28	0.21	0.04	0.17	0.15	0.29	0.06	1.00	0.25	0.20	0.17
9	0.06	0.03	0.00	0.00	0.20	0.04	0.00	0.25	1.00	0.26	0.38
10	0.06	0.15	0.01	0.09	0.04	0.20	0.00	0.20	0.26	1.00	0.12
11	0.00	0.00	0.00	0.01	0.18	0.03	0.01	0.17	0.38	0.12	1.00

Figure 7: Cosine similarities across all sentence pairs in a cluster of 11 sentences.

It is important to realize that the cosine matrix hides in itself an infinite number of graphs, for each

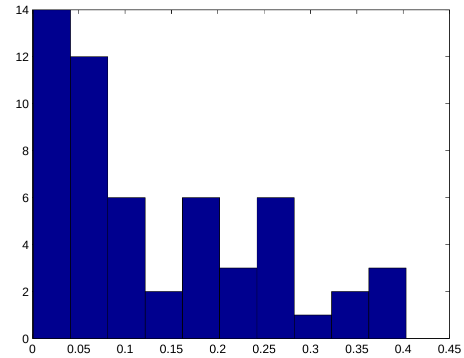

Figure 8: LexRank cosine histogram.

value of a cosine cutoff, t. This can be seen in the next two figures: Figures 9– 10. For example, if one lowers the threshold too much, the graph is almost fully connected. Conversely, raising the threshold eventually turns the graph into a set of disconnected components. The random walk is typically performed at the value of t at which approximately half of the node pairs are connected via edges.

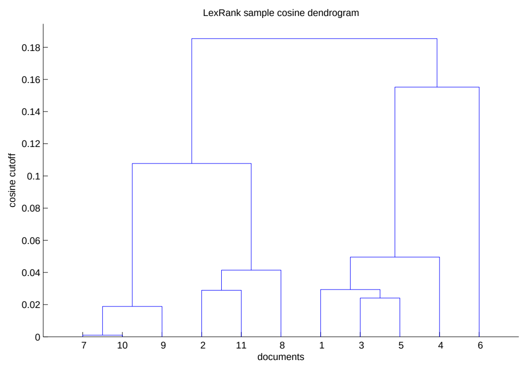

Figure 9: LexRank sample dendrogram.

Figure 11 shows the LexRank Java interface as used for text summarization.

4.2 Semi-supervised Passage Retrieval

Otterbacher et al. (Otterbacher et al., 2005) take (Erkan and Radev, 2004) one step further by introducing the concept of a *biased random walk* to address the problem of question-focused passage retrieval. In that problem the user issues a query in the form of a natural language question and expects to get a set of passages from the input documents

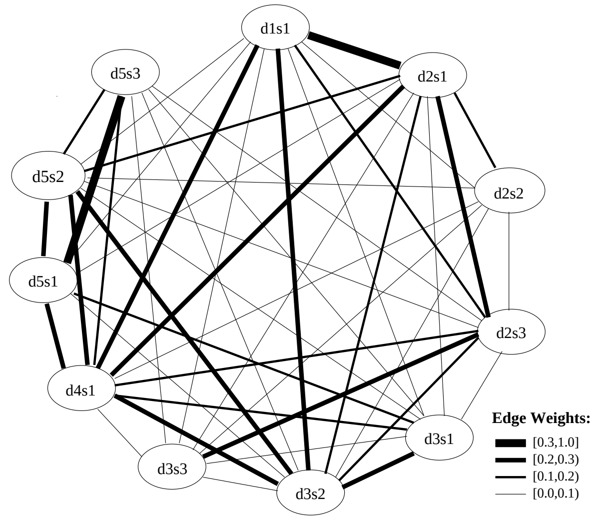

Figure 10: Weighted cosine similarity graph for the cluster in Figure 6.

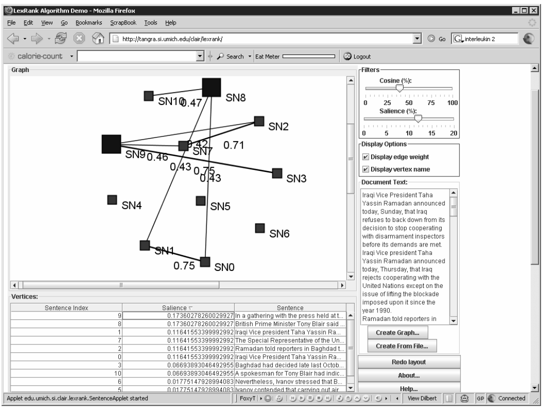

Figure 11: Lexrank interface.

that contain the answer to that question. The biased random walk is performed on a graph that is already seeded with known positive and negative examples. Then, each node is labeled in proportion to the percentage of times a random walk on the graph ends at that node. Given the presence of the initially labeled nodes, the nodes with the highest score eventually are the ones that are both similar to the seed nodes and are central to the document set. In other words, they are chosen eventually as the answer set by a mixture model that takes into account the known seeds (positive or negative) and the centrality score as in the previous section. The graph consists of both sentences (paragraphs) and features (content words th

ti e

ve

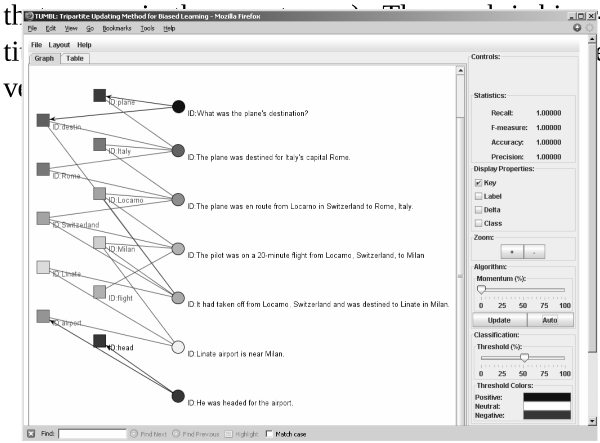

Figure 12: Biased lexrank as used for semi-supervised passage retrieval.

In the example shown in Figure 12, the top right-hand node is initially labeled as positive (dark) whereas the bottom right-hand node is labeled as negative (clear node). During the labeling (performed using the method of relaxations), each node's shadedness changes until the process converges. At the end, darker nodes are returned as relevant to the user question. Note that some of them contain no words in common with the original query.

4.3 Keyword Extraction

The task of a keyword extraction application is to automatically identify in a text a set of terms that best describe the document. Such keywords may constitute useful entries for building an automatic index for a document collection, can be used to classify a text, or may serve as a concise summary for a given document. A system for automatic identification of important terms can also be used for the problem of terminology extraction, and construction of domain-specific dictionaries. A random-walk algorithm for keyword extraction has been proposed in (Mihalcea and Tarau, 2004), where a graph is constructed on an input text by adding all the words in the text as nodes in the graph, and connecting them by a co-occurrence relation constrained by the distance between the words.

Figure 13 shows a sample graph built for a short scientific text. A random-walk run on such a graph of co-occurrences leads to a ranking over the importance of the words in the text; in a post-processing phase, words selected as important by the ranking algorithm and found next to each other in the text are collapsed into a single phrase. Interestingly, experiments comparing this ranking with the traditional *tf.idf* showed that the scores assigned by the random-walk can differ significantly. In fact, evaluations on a data set of scientific abstracts showed that the random-walk method is superior to the *tf.idf* method for keyword extraction, and it also improved over previously published state-of-the-art supervised methods for keyword extraction.

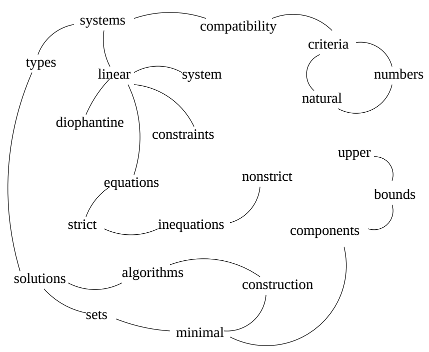

Figure 13: Sample graph built for keyword extraction.

4.4 Further Reading

A large bibliography appears on the first author's web site and also on w.textgraphs.org.

Acknowledgments

This paper is partially based on work done previously by the two authors. That work was partially funded by NSF grants IIS 0534323 "Collaborative Research: BlogoCenter - Infrastructure for Collecting, Mining and Accessing Blogs", IIS 0329043 "Probabilistic and link-based Methods for Exploiting Very Large Textual Repositories", BCS 0527513 "DHB: The Dynamics of Political Representation and Political Rhetoric", and by NIH grants R01 LM008106 "Representing and Acquiring Knowledge of Genome Regulation" and U54 DA021519 "National Center for Integrative Bioinformatics', all to Dragomir Radev. This work was also supported in part by a research grant #003594 on "Graph-based Natural Language Processing" from the Texas Advanced Research Program and by a Google grant on "Finding Important Information in Unstructured Text," both of them awarded to Rada Mihalcea.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the other sponsors.

References

Silvia M. G. Caldeira, Thierry C. Petit Lob ao, R. F. S. Andrade, Alexis Neme, and J. G. V. Miranda. 2006. The network of concepts in written texts. *European Physical Journal B*, 49(4):523–529, February.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336.

J. Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240.

Güneş Erkan and Dragomir Radev. 2004. The university of Michigan at duc 2004. In *Document Understanding Conference (DUC)*, Boston, Massachusetts, May.

A. Esuli and F. Sebastiani. 2007. PageRanking wordnet synsets: An application to opinion mining. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Ramon Ferrer-i-Cancho and Ricard V. Sole. 2001. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268(1482):2261–2265, November.

T. Hughes and D. Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of EMNLP 2007*, Prague, Czech Republic.

Y. J. and T. H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

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

A. P. Masucci and G. J. Rodgers. 2006. Network properties of written human language. *Physical Review E*, 74, August 2,.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

R. Mihalcea, P. Tarau, and E. Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20st International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Rada Mihalcea. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 411–418, Vancouver, British Columbia, Canada, October.

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India.

Cristina Nicolae and Gabriel Nicolae. 2006. Bestcut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 275–283, Sydney, Australia, July.

Z.Y. Niu, D.H. Ji, and C.L. Tan. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Michigan. Association for Computational Linguistics.

Jahna Otterbacher, Güneş Erkan, and Dragomir Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915–922, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 271–278, Barcelona, Spain, July.

Dragomir R. Radev, Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan. 2001. NewsInEssence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of Human Language Technology Conference (HLT 2001)*.

Dragomir R. Radev. 2004. Weakly supervised graph-based methods for classification. Technical Report CSE-TR-500-04, University of Michigan. Department of Electrical Engineering and Computer Science.

Gerard Salton, James Allan, Chris Buckley, and Amit Singhal. 1994. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264(5164):1421–1426.

Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. 33(2):193–207, March.

Mariano Sigman and Guillermo A. Cecchi. 2002. Global organization of the Wordnet lexicon. *Proceedings of the National Academy of Sciences of the United States of America*, 99(3):1742–1747, February 5,.

R. Sinha and R. Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007)*, Irvine, CA.

M. Steyvers and J.B. Tenenbaum. 2005. Graph theoretic analyses of semantic networks: Small worlds in semantic networks. *Cognitive Science*, 29:41–78.

Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *ICML*

'04: Proceedings of the twenty-first international conference on Machine learning, page 103, New York, NY, USA.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei.

J. Wiebe and R. Mihalcea. 2006. Word sense and subjectivity. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.

Xiaojin Zhu and John Lafferty. 2005. Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In Saso Dzeroski, Luc De Raedt, and Stefan Wrobel, editors, *Proceedings of the Twenty-Second International Conference on Machine Learning (ICML '05)*, Bonn, Germany, August 7-11,. ACM Press.