LexNet: A Graphical Environment for Graph-Based NLP

Dragomir R. Radev[1,2], **Güneş Erkan**[1], **Anthony Fader**[3],
Patrick Jordan[1], Siwei Shen[1], and James P. Sweeney[2]
Department of Electrical Engineering and Computer Science
School of Information
Department of Mathematics
University of Michigan
Ann Arbor, MI 48109
{radev, gerkan, afader, prjordan, shens, jpsweeney}@umich.edu

Abstract

This interactive presentation describes LexNet, a graphical environment for graph-based NLP developed at the University of Michigan. LexNet includes LexRank (for text summarization), biased LexRank (for passage retrieval), and TUMBL (for binary classification). All tools in the collection are based on random walks on *lexical graphs*, that is graphs where different NLP objects (e.g., sentences or phrases) are represented as nodes linked by edges proportional to the lexical similarity between the two nodes. We will demonstrate these tools on a variety of NLP tasks including summarization, question answering, and prepositional phrase attachment.

1 Introduction

We will present a series of graph-based tools for a variety of NLP tasks such as text summarization, passage retrieval, prepositional phrase attachment, and binary classification in general.

Recently proposed graph-based methods (Szummer and Jaakkola, 2001; Zhu and Ghahramani, 2002b; Zhu and Ghahramani, 2002a; Toutanova et al., 2004) are particularly well suited for transductive learning (Vapnik, 1998; Joachims, 1999). Transductive learning is based on the idea (Vapnik, 1998) that instead of splitting a learning problem into two possibly harder problems, namely induction and deduction, one can build a model that covers both labeled and unlabeled data. Unlabeled data are abundant as well as significantly cheaper than labeled data in a variety of natural language applications. Parsing and machine translation both offer examples of this relationship, with unparsed text from the Web and untranslated texts being computationally less costly. These can then be used to supplement manually translated and aligned corpora. Hence transductive methods are of great potential for NLP problems and, as a result, LexNet includes a number of transductive methods.

2 LexRank: text summarization

LexRank (Erkan and Radev, 2004) embodies the idea of representing a text (e.g., a document or a collection of related documents) as a graph. Each node corresponds to a sentence in the input and the edge between two nodes is related to the lexical similarity (either cosine similarity or n-gram generation probability) between the two sentences. LexRank computes the steady-state distribution of the random walk probabilities on this similarity graph. The LexRank score of each node gives the probability of a random walk ending up in that node in the long run. An extractive summary is generated by retrieving the sentences with the highest score in the graph. Such sentences typically correspond to the nodes that have strong connections to other nodes with high scores in the graph. Figure 1 demonstrates LexRank.

3 Biased LexRank: passage retrieval

The basic idea behind Biased LexRank is to label a small number of sentences (or passages) that are relevant to a particular query and then propagate relevance from these sentences to other (unannotated) sentences. Relevance propagation is performed on a bipartite graph. In that graph, one of the modes corresponds to the sentences and the other – to certain words from these sentences. Each sentence is connected to the words that appear in it. Thus indirectly, each sentence is two hops away from any other sentence that shares words in it. Intuitively, the sentences that are close to the labeled sentences tend to get higher scores. However, the relevance propagation en-

Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pages 45–48,
Sydney, July 2006.

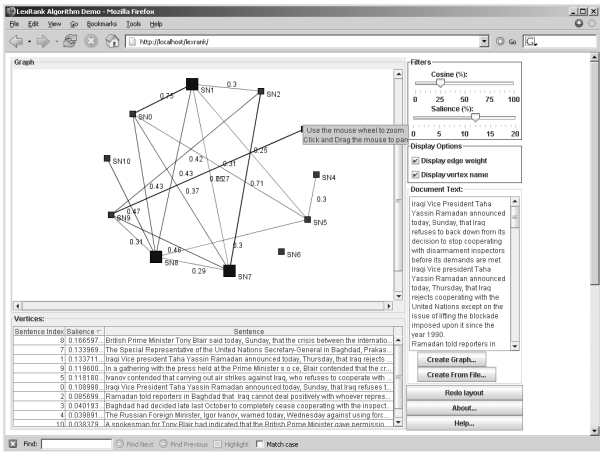


Figure 1: A sample snapshot of LexRank. A 3-sentence summary is produced from a set of 11 related input sentences. The summary sentences are shown as larger squares.

ables us to mark certain sentences that are not immediate neighbors of the labeled sentences via indirect connections. The effect of the propagation is discounted by a parameter at each step so that the relationships between closer nodes are favored more. Biased LexRank also allows for negative relevance to be propagated through the network as the example shows. See Figures 2– 3 for a demonstration of Biased LexRank.

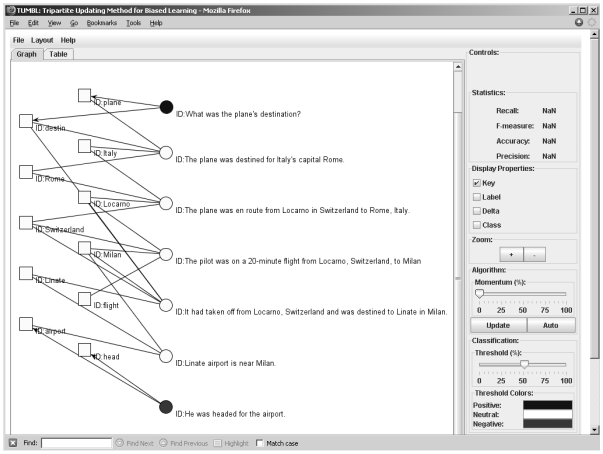


Figure 2: Display of Biased LexRank. One sentence at the top is annotated as positive while another at the bottom is marked negative. Sentences are displayed as circles and the word features are shown as squares.

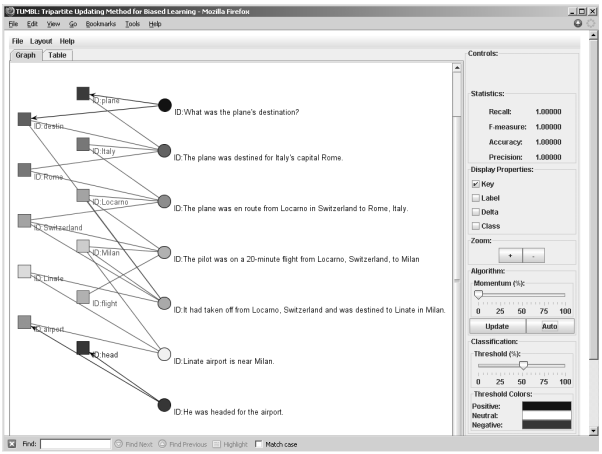


Figure 3: After convergence of Biased LexRank.

4 TUMBL: prepositional phrase attachment

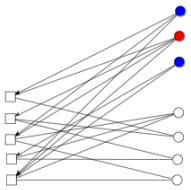
A number of NLP problems such as word sense disambiguation, text categorization, and extractive summarization can be cast as classification problems. This fact is used to great effect in the design and application of many machine learning methods used in modern NLP, including TUMBL, through the utilization of vector representations. Each object is represented as a vector $\mathbf{x}$ of features. The main assumption made is that a pair of objects $\mathbf{x}$ and $\mathbf{y}$ will be classified the same way if the distance between them in some space D is small (Zhu and Ghahramani, 2002a).

This algorithm propagates polarity information first from the labeled data to the features, capturing whether each feature is more indicative of positive class or more negative learned. Such information is further transferred to the unlabeled set. The backward steps update feature polarity with information learned from the structure of the unlabeled data. This process is repeated with a damping factor to discount later rounds. This process is illustracted in Figure 4. TUMBL was first described in (Radev, 2004). A series of snapshots showing TUMBL in Figures 5– 7.

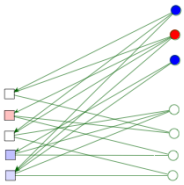
5 Technical information

5.1 Code implementation

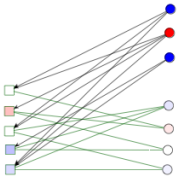
The LexRank and TUMBL demonstrations are provided as both an applet and an application. The user is presented with a graphical visualization of the algorithm that was conveniently developed using the JUNG API (`http://jung.sourceforge.net/faq.html`).



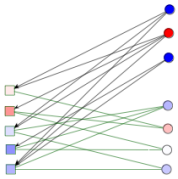
(a) Initial graph



(b) Forward pass



(c) Backward pass



(d) Convergence

Figure 4: TUMBL snapshots: the circular vertices are objects while the square vertices are features. (a) The initial graph with features showing no bias. (b) The forward pass where objects propagate labels forward. (c) The backward pass where features propagate labels backward. (d) Convergence of the TUMBL algorithm after successive iterations.

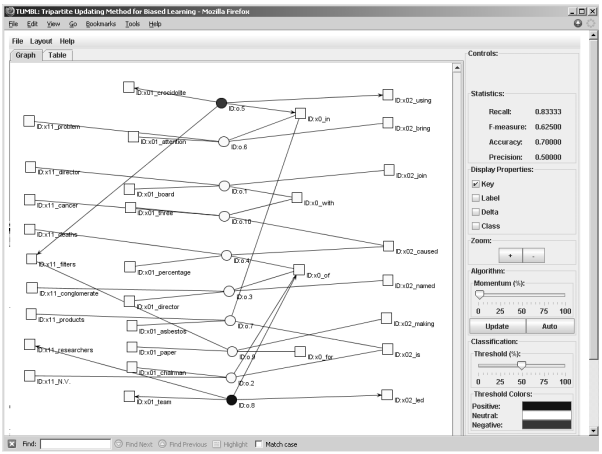


Figure 5: A 10-pp prepositional phrase attachment problem is displayed. The head of each prepositional phrase is ine middle column. Four types of features are represented in four columns. The first column is Noun1 in the 4-tuple. The second column is Noun2. The first column from the right is verb of the 4-tuple while the second column from the right is the actual head of the prepositional phrase. At this time one positive and one negative example (high and low attachment) are annotated. The rest of the circles correspond to the unlabeled examples.

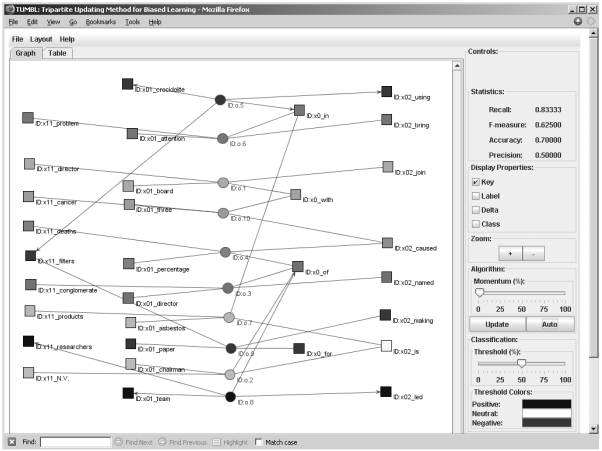


Figure 6: The final configuration.

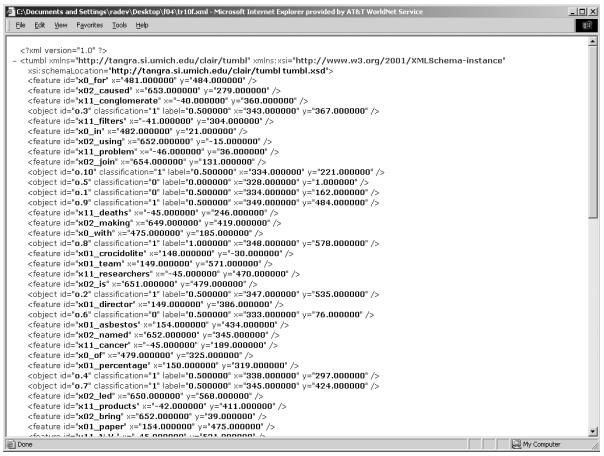


Figure 7: XML file corresponding to the PP attachment problem. The XML DTD allows layout information to be encoded along with algorithmic information such as label and polarity.

In TUMBL, each object is represented by a circular vertex in the graph and each feature as a square. Vertices are assigned a color according to their label. The colors are assignable by the user and designate the probability of membership of a class.

To allow for a range of uses, data can be entered either though the GUI or read in from an XML file. The schema for TUMBL files is shown at `http://tangra.si.umich.edu/clair/tumbl`.

In the LexRank demo, each sentence becomes a node. Selected nodes for the summary are shown in larger size and in blue while the rest are smaller and drawn in red. The link between two nodes has a weight proportional to the lexical similarity between the two corresponding sentences. The demo also reports the metrics precision, recall, and F-measure.

5.2 Availability

The demos are available both as locally based and remotely accessible from `http://tangra.si.umich.edu/clair/lexrank` and `http://tangra.si.umich.edu/clair/tumbl`.

6 Acknowledgments

This work was partially supported by the U.S. National Science Foundation under the following two grants: 0329043 “Probabilistic and link-based Methods for Exploiting Very Large Textual Repositories” administered through the IDM program and 0308024 “Collaborative Research: Semantic Entity and Relation Extraction from Web-Scale Text Document Collections” run by the HLC program. All opinions, findings, conclusions, and recommendations in this paper are made by the authors and do not necessarily reflect the views of the National Science Foundation.

References

Güneş Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *ICML ’99*.

Dragomir Radev. 2004. Weakly supervised graph-based methods for classification. Technical Report CSE-TR-500-04, University of Michigan.

Martin Szummer and Tommi Jaakkola. 2001. Partially labeled classification with Markov random walks. In *NIPS ’01*, volume 14. MIT Pres.

Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *ICML ’04*, New York, New York, USA. ACM Press.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Xiaojin Zhu and Zoubin Ghahramani. 2002a. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107.

Xiaojin Zhu and Zoubin Ghahramani. 2002b. Towards semi-supervised classification with Markov random fields. Technical report, CMU-CALD-02-106.