Graph-Based Text Representation for Novelty Detection

Michael Gamon
Microsoft Research
Redmond, WA 98052
mgamon@microsoft.com

Abstract

We discuss several feature sets for novelty detection at the sentence level, using the data and procedure established in task 2 of the TREC 2004 novelty track. In particular, we investigate feature sets derived from graph representations of sentences and sets of sentences. We show that a highly connected graph produced by using sentence-level term distances and pointwise mutual information can serve as a source to extract features for novelty detection. We compare several feature sets based on such a graph representation. These feature sets allow us to increase the accuracy of an initial novelty classifier which is based on a bag-of-word representation and KL divergence. The final result ties with the best system at TREC 2004.

1 Introduction

Novelty detection is the task of identifying novel information given a set of already accumulated background information. Potential applications of novelty detection systems are abundant, given the "information overload" in email, web content etc. Gabrilovich et al (2004), for example, describe a scenario in which a newsfeed is personalized based on a measure of information novelty: the user can be presented with pieces of information that are novel, given the documents that have already been reviewed. This will spare the user the task of sifting through vast amounts of duplicate and redundant information on a topic to find bits and pieces of information that are of interest.

In 2002 TREC introduced a novelty track (Harman 2002), which continued — with major changes — in 2003 (Soboroff and Harman 2003) and 2004 (Voorhees 2004). In 2002 the task was to identify the set of relevant and novel sentences from an ordered set of documents within a TREC topic. Novelty was defined as "providing new information that has not been found in any previously picked sentences". Relevance was defined as "relevant to the question or request made in the description section of the topic". Inter-annotator agreement was low (Harman 2002). There were 50 topics for the novelty task in 2002. For the 2003 novelty track a number of major changes were made. Relevance and novelty detection were separated into different tasks, allowing a separate evaluation of relevance detection and novelty detection. In the 2002 track, the data proved to be problematic since the percentage of relevant sentences in the documents was small. This, in turn, led to a very high percentage of relevant sentences being novel, given that amongst the small set of relevant sentences there was little redundancy. 50 new topics were created for the 2003 task, with a better balance of relevant and novel sentences. Slightly more than half of the topics dealt with "events," the rest with "opinions."

The 2004 track used the same tasks, the same number of topics and the same split between event and opinion topics as the 2003 track.

For the purpose of this paper, we are only concerned with novelty detection, specifically with task 2 of the 2004 novelty track, as described in more detail in the following section.

The question that we investigate here is: what is a meaningful feature set for text representation for novelty detection? This is obviously a far-reaching and loaded question. Possibilities range from simple bag-of-word features to features derived from sophisticated linguistic representations. Ultimately, the question is open-ended since there will always be another feature or feature combination that could/should be exploited. For our experiments, we have decided to focus more narrowly on the usefulness of features derived from graph representations and we have restricted ourselves to representations that do not require linguistic analysis. Simple bag-of-word metrics like KL divergence establish a baseline for classifier performance. More sophisticated metrics can be defined on the basis of graph representations. Graph representations of text can be constructed without performing linguistic analysis, by using term distances in sentences and pointwise mutual information between terms to form edges between term-vertices. A term-distance based representation has been used successfully for a variety of tasks in Mihalcea (2004) and Mihalcea and Tarau (2004).

2 Previous work

There were 13 participants and 54 submitted runs for the 2004 TREC novelty track task 2. Each participant submitted up to five runs with different system configurations. Metrics and approaches varied widely, from purely string based approaches to systems that used sophisticated linguistic components for synonymy resolution, coreference resolution and named entity recognition. Many systems employed a thresholding approach to the task, defining a novelty metric and then determining a sentence to be novel if the threshold is exceeded (e.g. Blott et al. 2004, Zhang et al. 2004, Abdul-Jaleel et al. 2004, Eichmann et al. 2004, Erkan 2004). Thresholds are either determined on the 2003 data, are based on a notion of mean score, or are determined in an ad hoc manner[1]. Tomiyama et al (2004), similar to our approach, use an SVM classifier to make the binary classification of a sentence as novel or not.

The baseline result for the 2004 task 2 was an average F-measure of 0.577. This baseline is achieved if all relevant sentences are categorized as novel. The difficulty of the novelty detection task is evident from the relatively low score achieved by even the best systems. The five best-performing runs were:

1. Blott et al. (2004) (Dublin City University): using a tf.idf based metric of "importance value" at an ad hoc threshold: **0.622.**
2. Tomiyama et al. (2004) (Meiji University): using an SVM classifier trained on 2003 data, features based on conceptual fuzzy sets derived from a background corpus: **0.619**.
3. Abdul-Jaleel et al. (2004) (UMass): using named entity recognition, using cosine similarity as a metric and thresholds derived from the 2003 data set: **0.618**.
4. Schiffman and McKeown (2004) (Columbia): using a combination of tests based on weights (derived from a background corpus) for previously unseen words with parameters trained on the 2003 data set, and taking into account the novelty status of the previous sentence: **0.617**.
5. Tomiyama et al (2004) (Meiji University): slight variation of the system described above, with one of the features (scarcity measure) eliminated: **0.617**.

As this list shows, there was no clear tendency of any particular kind of approach outperforming others. Among the above four systems and five runs, there are thresholding and classification approaches, systems that use background corpora and conceptual analysis and systems that do not.

3 Experimental setup

3.1 The task

Task 2 of the 2004 novelty track is formulated as follows:

> Task 2: Given the relevant sentences in the complete document set (for a given topic), identify all novel sentences.

The procedure is sequential on an ordered list of sentences per topic. For each Sentence S_i the determination needs to be made whether it is novel given the previously seen sentences S_1 through S_{i-1}.

[1] Unfortunately, some of the system descriptions are unclear about the exact rationale for choosing a particular threshold.

The evaluation metric for the novelty track is F_1-measure, averaged over all 50 topics.

3.2 Novelty detection as classification

For the purpose of this paper we view novelty detection as a supervised classification task. While the supervised approach has its limitations in real-life scenarios where annotated data are hard to come by, it can serve as a testing ground for the question we are interested in: the evaluation of feature sets and text representations.

At training time, a feature vector is created for each tagged sentence S and the set of sentences that comprise the already seen information that S is compared to. Features in the vector can be features of the tagged sentence, features of the set of sentences comprising the given background information and features that capture a relation between the tagged sentence and the set of background sentences. A classifier is trained on the set of resulting feature vectors. At evaluation time, a feature vector is extracted from the sentence to be evaluated and from the set of sentences that form the background knowledge. The classifier then determines whether, given the feature values of that vector, the sentence is more likely to be novel or not.

We use the TREC 2003 data set for training, since it is close to the 2004 data set in its makeup. We train Support Vector Machines (SVMs) on the 2003 data, using the LibSVM tool (Chang and Lin 2001). Following the methodology outlined in Chang and Lin 2003, we use radial basis function (RBF) kernels and perform a grid search on two-fold cross validated results on the training set to identify optimal parameter settings for the penalty parameter C and the RBF parameter γ. Continuously valued features are scaled to values between -1 and 1. The scaling range is determined on the training set and the same range is applied to the test set.

The text was minimally preprocessed before extracting features: stop words were removed, tokens were lowercased and punctuation was stripped from the strings.

4 Text representations and features

4.1 KL divergence as a feature

Treating sentences as an unordered collection of terms, the information-theoretic metric of KL divergence (or relative entropy) has been successfully used to measure "distance" between documents by simply comparing the term distributions in a document compared to another document or set of documents. The notions of distance and novelty are closely related: if a new document is very distant from the collection of documents that has been seen previously, it is likely to contain new, previously unseen information. Gabrilovich et al. (2004), for example, report on a successful use of KL divergence for novelty detection. KL divergence is defined in Equation 1:

$$\sum_w p_d(w) \log \frac{p_d(w)}{p_R(w)}$$

Equation 1: KL divergence.

w belongs to the set of words that are shared between document *d* and document (set) *R*. p_d and p_R are the probability distributions of words in *d* and *R*, respectively. Both $p_d(w)$ and $p_R(w)$ need to be non-zero in the equation above. We used simple add-one smoothing to ensure non-zero values. While it is conceivable that KL divergence could take into account other features than just bag-of-words information, we restrict ourselves to this particular use of the measure since it corresponds to the typical use in novelty detection.

4.2 Term distance graphs: from text to graph without linguistic analysis

KL divergence as described above treats a document or sentence as an unordered collection of words. Language obviously provides more structure than that. Linguistic resources can impose structure on a string of words through consultation of linguistic knowledge (either hand-coded or learned from a tagged corpus). Even without any outside knowledge, however, the order of words in a sentence provides a means to construct a highly connected undirected graph with the words as vertices. The intuition here is:

1. All words in a sentence have some relationship to all other words in the sentence, modulo a "window size" outside of which the relationship is not taken into consideration
2. The closer two words are to each other, the stronger their connection tends to be[2]

It follows from (2) that weights on the edges will be inversely proportional to the distance between two words (vertices). In the remainder of the paper we will refer to these graphs as TD (term distance) graphs. Of course (1) and (2) are rough generalizations with many counterexamples, but without the luxury of linguistic analysis this seems to be a reasonable step to advance beyond simple bag-of-word assumptions. Multiple sentence graphs can then be combined into a highly connected graph to represent text. Mihalcea (2004) and Mihalcea and Tarau (2004) have successfully explored very similar graph representations for extractive summarization and key word extraction.

In addition to distance, we also employ pointwise mutual information as defined in Equation 2 between two words/vertices to enter into the calculation of edge weight[3]. This combination of distance and a cooccurrence measure such as PMI is reminiscent of decaying language models, as described for IR, for example, in Gao et al. (2002)[4]. Cooccurrence is counted at the sentence level, i.e. $P(i, j)$ is estimated by the number of sentences that contain both terms w_i and w_j, and $P(i)$ and $P(j)$ are estimated by counting the total sentences containing w_i and w_j, respectively. As the set of seen sentences grows and cooccurrence between words becomes more prevalent, PMI becomes more influential on edge weights, strengthening edges between words that have high PMI.

$$PMI_{(i,j)} = \log_2 \frac{P(i, j)}{P(i)P(j)}$$

Equation 2: Pointwise Mutual Information (PMI) between two terms *i* and *j*.

Formally, the weight *wt* for each edge in the graph is defined as in Equation 3, where $d_{i,j}$ is the distance between words w_i and w_j.and PMI(i,j) is the pointwise mutual information between words w_i and w_j, given the sentences seen so far. For the purpose of Equation 3 we ignored negative PMI values, i.e. we treated negative PMI values as 0.

$$wt_{i,j} = \frac{1 + PMI(i, j)}{d_{i,j}^2}$$

Equation 3: Assigning weight to an edge between two vertices.

We imposed a "window size" as a limit on the maximum distance between two words to enter an edge relationship. Window size was varied between 3 and 8; on the training set a window size of 6 proved to be optimal.

On a TD graph representation, we can calculate various features based on the strengths and number of connections between words. In novelty detection, we can model the growing store of background information by adding each "incoming" sentence graph to the existing background graph. If an "incoming" edge already exists in the background graph, the weight of the "incoming" edge is added to the existing edge weight.

Figure 1 shows a subset of a TD graph for the first two sentences of topic N57. The visualization is generated by the Pajek tool (Bagatelj and Mrvar).

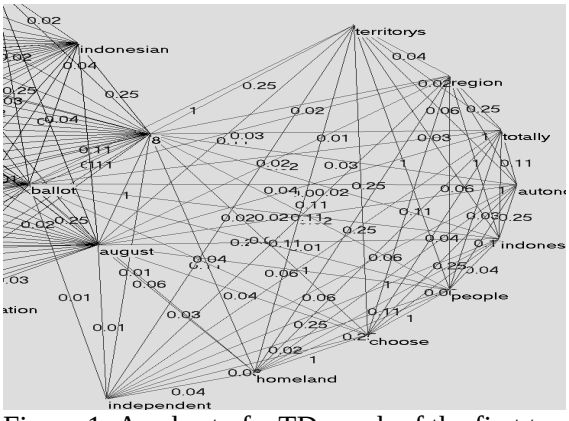

Figure 1: A subset of a TD graph of the first two sentences of topic N57.

[2] This view is supported by examining dependency structures derived from the Penn Tree Bank and mapping the probability of a dependency to the distance between words. See also Eisner and Smith (2005) who explore this generalization for dependency parsing.

[3] We also computed results from a graph where the edge weight is determined only by term distance, without PMI. These results were consistently worse than the ones reported here.

[4] We are grateful to an anonymous reviewer for pointing this out.

4.3 Graph features

4.3.1 Simple Graph features

In novelty detection, graph based features allow to assess the change a graph undergoes through the addition of a new sentence. The intuition behind these features is that the more a graph changes when a sentence is added, the more likely the added sentence is to contain novel information. After all, novel information may be conveyed even if the terms involved are not novel. Establishing a new relation (i.e. edge in the graph) between two previously seen terms would have exactly that effect: old terms conveying new information. KL divergence or any other measure of distributional similarity is not suited to capture this scenario. As an example consider a news story thread about a crime. The various sentences in the background information may mention the victim, multiple suspects, previous convictions, similar crimes etc. When a new sentence is encountered where one suspect's name is mentioned in the same sentence with the victim, at a close distance, none of these two terms are new. The fact that suspect and victim are mentioned in one sentence, however, may indicate a piece of novel information: a close relationship between the two that did not exist in the background story.

We designed 21 graph based features, based on the following definitions:

- *Background graph:* the graph representing the previously seen sentences.
- *G(S):* the graph of the sentence that is currently being evaluated.
- *Reinforced background edge:* an edge that exists both in the background graph and in G(*S)*.
- *Added background edge:* a new edge in G(S) that connects two vertices that already exist in the background graph.
- *New edge:* an edge in G(S) that connects two previously unseen vertices.
- *Connecting edge:* an edge in G(S) between a previously unseen vertex and a previously seen vertex.

The 21 features are:

- number of new edges
- number of added background edges
- number of background edges
- number of background vertices
- number of connecting edges
- sum of weights on new edges
- sum of weights on added background edges
- sum of weights on connecting edges
- background connectivity (ratio between edges and vertices)
- connectivity added by *S*
- ratio between added background edges and new edges
- ratio between new edges and connecting edges
- ratio between added background edges and connecting edges
- ratio between the sum of weights on new edges and the sum of weights on added background edges
- ratio between the sum of weights on new edges and the sum of weights on connecting edges
- ratio between the sum of weights on added background edges and the sum of weights on connecting edges
- ratio between sum of weights on added background edges and the sum of pre-existing weights on those edges
- ratio between sum of weights on new edges and sum of weight on background edges
- ratio between sum of weights added to reinforced background edges and sum of background weights
- ratio between number of added background edges and reinforced background edges
- number of background edges leading from those background vertices that have been connected to new vertices by G(*S)*

We refer to this set of 21 features as *simple graph features*, to distinguish them from a second set of graph-based features that are based on TextRank.

4.3.2 TextRank features

The TextRank metric, as described in Mihalcea and Tarau (2004) is inspired by the PageRank

metric which is used for web page ranking[5]. TextRank is designed to work well in text graph representations: it can take edge weights into account and it works on undirected graphs. TextRank calculates a weight for each vertex, based on Equation 4.

$$TR(V_i) = (1-d) + d * \left(\sum_{V_j \in NB(V_i)} \frac{wt_{ji}}{\sum_{V_k \in NB(V_j)} wt_{jk}} TR(V_j) \right)$$

Equation 4: The TextRank metric.

where $TR(V_i)$ is the TextRank score for vertex i, $NB(V_i)$ is the set of neighbors of V_i, i.e. the set of nodes connected to V_i by a single edge, wt_{xy} is the weight of the edge between vertex x and vertex y, and d is a constant "dampening factor", set at 0.85[6]. To calculate TR, an initial score of 1 is assigned to all vertices, and the formula is applied iteratively until the difference in scores between iterations falls below a threshold of 0.0001 for all vertices (as in Mihalcea and Tarau 2004).

The TextRank score itself is not particularly enlightening for novelty detection. It measures the "importance" rather than the novelty of a vertex - hence its usefulness in keyword extraction. We can, however, derive a number of features from the TextRank scores that measure the change in scores as a result of adding a sentence to the graph of the background information. The rationale is that the more the TextRank scores are "disturbed" by the addition of a new sentence, the more likely it is that the new sentence carries novel information. We normalize the TextRank scores by the number of vertices to obtain a probability distribution. The features we define on the basis of the (normalized) TextRank metric are:

1. sum of TR scores on the nodes of S, after adding S
2. maximum TR score on any nodes of S
3. maximum TR score on any background node before adding S
4. delta between 2 and 3
5. sum of TR scores on the background nodes (after adding S)
6. delta between 5 and 1
7. variance of the TR scores before adding S
8. variance of TR scores after adding S
9. delta between 7 and 8
10. ratio of 1 to 5
11. KL divergence between the TR scores before and after adding S

5 Results

To establish a baseline, we used a simple bag-of-words approach and KL divergence as a feature for classification. Employing the protocol described above, i.e. training the classifier on the 2003 data set, and optimizing the parameters on 2 folds of the training data, we achieve a surprisingly high result of 0.618 average F-measure on the 2004 data. This result would place the run at a tie for third place with the UMass system in the 2004 competition.

In the tables below, *KL* refers to the KL divergence feature, *TR* to the TextRank based features and *SG* to the simple graph based features.

Given that the feature sets we investigate possibly capture orthogonal properties, we were also interested in using combinations of the three feature sets. For the graph based features we determined on the training set that results were optimal at a "window size" of 6, i.e. if graph edges are produced only if the distance between terms is six tokens or less. All results are tabulated in Table 1, with the best results boldfaced.

Feature set	Average F measure
KL	0.618
TR	0.600
SG	0.619
KL + SG	**0.622**
KL + SG + TR	**0.621**
SG + TR	0.615
TR + KL	0.618

Table 1: Performance of the different feature sets.

We used the McNemar test to determine pairwise statistical significance levels between the novelty classifiers based on different feature sets[7]. The two (boldfaced) best results from Table 1 are significantly different from the baseline at 0.999 confidence. Individual sentence level

[5] Erkan and Radev (2005) introduced *LexRank* where a graph representation of a set of sentences is derived from the cosine similarity between sentences. Kurland and Lee (2004) derive a graph representation for a set of documents by linking documents X and Y with edges weighted by the score that a language model trained on X assigns to Y.

[6] Following Mihalcea and Tarau (2004), who in turn base their default setting on Brin and Page (1998).

[7] We could not use the Wilcoxon rank test for our results since we only had binary classification results for each sentence, as opposed to individual (class probability) scores.

classifications from the official 2004 runs were not available to us, so we were not able to test for statistical significance on our results versus TREC results.

6 Summary and Conclusion

We showed that using KL divergence as a feature for novelty classification establishes a surprisingly good result at an average F-measure of 0.618, which would top all but 3 of the 54 runs submitted for task 2 in the TREC novelty track in 2004. To improve on this baseline we computed graph features from a highly connected graph built from sentence-level term cooccurrences with edges weighted by distance and pointwise mutual information. A set of 21 "simple graph features" extracted directly from the graph perform slightly better than KL divergence, at 0.619 average F-measure. We also computed TextRank features from the same graph representation. TextRank features by themselves achieve 0.600 average F-measure. The best result is achieved by combining feature sets: Using a combination of KL features and simple graph features produces an average F-measure of 0.622.

Being able to establish a very high baseline with just the use of KL divergence as a feature was surprising to us: it involves a minimal approach to novelty detection. We believe that the high baseline indicates that a classification approach to novelty detection is promising. This is corroborated by the very good performance of the runs from Meiji University which also used a classifier.

The second result, i.e. the benefit obtained by using graph based features was in line with our expectations. It is a reasonable assumption that the graph features would be able to add to the information that a feature like KL divergence can capture. The gains were statistically significant but very modest, which poses a number of questions. First, our feature engineering may be less than optimal, missing important information from a graph-based representation. Second, the classification approach may be suffering from inherent differences between the training data (TREC 2003) and the test data (TREC 2004). To explore this hypothesis, we trained SVMs on the KL + SG feature set with default settings on three random folds of the 2003 and 2004 data. For these experiments we simply measured accuracy. The baseline accuracy (predicting the majority class label) was 65.77% for the 2003 data and 58.59% for the 2004 data. Average accuracy for the threefold crossvalidation on 2003 data was 75.72%, on the 2004 data it was 64.88%. Using the SVMs trained on the 2003 data on the three folds of the 2004 data performed below baseline at 55.07%. These findings indicate that the 2003 data are indeed not an ideal fit as training material for the 2004 task.

With these results indicating that graph features can be useful for novelty detection, the question becomes which graph representation is best suited to extract these features from. A highly connected term-distance based graph representation, with the addition of pointwise mutual information, is a computationally relatively cheap approach. There are at least two alternative graph representations that are worth exploring.

First, a "true" dependency graph that is based on linguistic analysis would provide a less connected alternative. Such a graph would, however, contain more information in the form of directed edges and edge labels (labels of semantic relations) that could prove useful for novelty detection. On the downside, it would necessarily be prone to errors and domain specificity in the linguistic analysis process.

Second, one could use the parse matrix of a statistical dependency parser to create the graph representation. This would yield a dependency graph that has more edges than those coming from a "1-best" dependency parse. In addition, the weights on the edges could be based on dependency probability estimates, and analysis errors would not be as detrimental since several alternative analyses enter into the graph representations.

It is beyond the scope of this paper to present a thorough comparison between these different graph representations. However, we were able to demonstrate that a computationally simple graph representation, which is based solely on pointwise mutual information and term distance, allows us to successfully extract useful features for novelty detection. The results that can be achieved in this manner only present a modest gain over a simple approach using KL divergence as a classification feature. The best achieved result, however, would tie for first place in the 2004 TREC novelty track,

in comparison to many systems which relied on relatively heavy analysis machinery and additional data resources.

References

Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Vladimir Batagelj, Andrej Mrvar: Pajek - Program for Large Network Analysis. Home page http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

Stephen Blott, Oisin Boydell, Fabrice Camous, Paul Ferguson, Georgina Gaughan, Cathal Gurrin, Gareth J. F. Jones, Noel Murphy, Noel O'Connor, Alan F. Smeaton, Barry Smyth, Peter Wilkins. 2004. Experiments in Terabyte Searching, Genomic Retrieval and Novelty Detection for TREC-2004. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems 30: 107-117.*

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chih-Chung Chang and Chih-Jen Lin. 2003. A Practical Guide to Support Vector Classification.

David Eichmann, Yi Zhang, Shannon Bradshaw, Xin Ying Qiu, Li Zhou, Padmini Srinivasan, Aditya Kumar Sehgal and Hudon Wong. 2004. Novelty, Question Answering and Genomics: The University of Iowa Response. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Jason Eisner and Noah A. Smith. 2005. Parsing with Soft and Hard Constraints on Dependency Length. *Proceedings of the International Workshop on Parsing Technologies (IWPT).*

Güneş Erkan. 2004. The University of Michigan in Novelty 2004. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Güneş Erkan and Dragomir Radev. 2004. LexRank: Graph-based Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research 22, pp. 457-479.*

Evgeniy Gabrilovich, Susan Dumais and Eric Horvitz. 2004. Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty. WWW13, 2004.

Jianfeng Gao, Jian-Yun Nie, Hongzhao He, Weijun Chena and Ming Zhou. 2002. Resolving Query Translation Ambiguity using a Decaying Co-occurrence Model and Syntactic Dependency Relations. Proceedings of SIGIR 2002, 183-190.

Donna Harman. 2002. Overview of the TREC 2002 Novelty Track. *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference (TREC 2002).*

Oren Kurland and Lillian Lee. 2004. PageRank without hyperlinks: structural re-ranking using links induced by language models. *Proceedings of SIGIR 2005, pp. 306-313.*

Rada Mihalcea. 2004. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, companion volume (ACL 2004).*

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004).*

Barry Schiffman and Kathleen R. McKeown. 2004. Columbia University in the Novelty Track at TREC 2004. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Ian Soboroff and Donna Harman. 2003. Overview of the TREC 2003 Novelty Task. *NIST Special Publication 500-255: The Twelfth Text REtrieval Conference (TREC 2003).*

Tomoe Tomiyama, Kosuke Karoji, Takeshi Kondo, Yuichi Kakuta and Tomohiro Takagi. 2004. Meiji University Web, Novelty and Genomics Track Experiments. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Ellen M. Voorhees. 2004. Overview of TREC 2004. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*

Hua-Ping Zhang, Hong-Bo Xu, Shuo Bai, Bin Wang and Xue-Qi Cheng. 2004. Experiments in TREC 2004 Novelty Track at CAS-ICT. *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004).*