Keyword Extraction and Headline Generation Using Novel Word Features

Songhua Xu[‡,§] **Shaohui Yang**[§] **Francis C.M. Lau**[§]

‡ Department of Computer Science, Yale University, New Haven, Connecticut, USA, 06520-8285
§ Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China

Abstract

We introduce several novel word features for keyword extraction and headline generation. These new word features are derived according to the background knowledge of a document as supplied by Wikipedia. Given a document, to acquire its background knowledge from Wikipedia, we first generate a query for searching the Wikipedia corpus based on the key facts present in the document. We then use the query to find articles in the Wikipedia corpus that are closely related to the contents of the document. With the Wikipedia search result article set, we extract the inlink, outlink, category and infobox information in each article to derive a set of novel word features which reflect the document's background knowledge. These newly introduced word features offer valuable indications on individual words' importance in the input document. They serve as nice complements to the traditional word features derivable from explicit information of a document. In addition, we also introduce a word-document fitness feature to characterize the influence of a document's genre on the keyword extraction and headline generation process. We study the effectiveness of these novel word features for keyword extraction and headline generation by experiments and have obtained very encouraging results.

Introduction

Being increasingly exposed to more and more information on the Internet, people of today have to be more selective about what to read. Keywords and headlines offer two important clues that can help a user quickly decide whether to skip, to scan, or to read the article. This paper addresses the problem of automatic keyword extraction and headline generation using novel word features.

Our keyword extraction method tries to identify the most important words in a document. This is known as an extractive approach. For headline generation, other than extractive approaches, there are also abstractive approaches (R.Soricut and D.Marcu 2007). Extractive approaches first identify the most important sentences in the document and then perform sentence compression to meet the length requirement for a headline. Abstractive approaches identify a list of important words or phrases in the document and then glue them together to create a headline text. Our headline generation method follows an abstractive approach. In both keyword extraction and headline generation, the extraction of keywords from a document is a core step.

The key contribution of our method for keyword extraction and headline generation is the introduction of several novel word features from observing a document's background knowledge which is derived through Wikipedia. To retrieve background knowledge related to a document, we first form a Wikipedia search query according to key facts present in the document. We then search the Wikipedia XML corpus (Denoyer and Gallinari 2006). Once a set of Wikipedia articles on the document's background knowledge are obtained, we can derive word features based on the inlink, outlink, category and infobox information in the retrieved articles. Previous work has explored the use of the link information in Wikipedia (Grineva, Grinev, and Lizorkin 2009; Mihalcea and Csomai 2007) for keyword extraction. The category and infobox information in Wikipedia, however, has never been used previously for keyword extraction. Our approach utilizes all four types of information during the keyword selection process. The results of our experiments confirm the effectiveness of our new word features for keyword extraction and headline generation by considering the background knowledge of a document.

The second contribution of our work is that when deriving word features for keyword extraction, we also consider the influence of the genre of a document. Previous studies have shown the effectiveness of observing document genres in document summarization (Stewart 2008; Dewdney, VanEss-Dykema, and MacMillan 2001). Similarly, we introduce a word-document genre fitness feature to characterize the likelihood of a word to be extracted as the document's keyword or headline word according to the word choice preference of the genre. Our experiments also have proved the effectiveness of this word-document genre fitness feature for keyword extraction and headline generation.

Related Work

Keyword Extraction

Traditional keyword extraction methods only use information explicitly contained in a document such as word fre-

quency and word position. In (Salton and Buckley 1987), a simple approach based on word frequency is proposed for keyword extraction. The TextRank algorithm introduced in (Mihalcea and Tarau 2004) uses three statistical properties including $tf \times idf$, distance, and key phrase frequency. The method proposed in (Zhang et al. 2006) uses $tf \times idf$, word position, POS of a word as well as the linkage between adjacent words as word features for keyword extraction. In (Ercan and Cicekli 2007), lexical chain features are used.

Recently, people have started to use Wikipedia for keyword extraction. Most closely related to our work here is the project called "Wikify!" (Mihalcea and Csomai 2007), which uses the link structure of Wikipedia to derive a novel word feature for keyword extraction. Grineva, Grinev, and Lizorkin (2009) utilized article titles and the link structure of Wikipedia to construct a semantic graph for keyword extraction. Unlike their approach, when we extract keywords, we use not only information explicitly contained in the document such as word frequency, position, and length but also the background knowledge of the document, which is acquired from Wikipedia via analyzing the inlink, outlink, category, and infobox information of the document's related articles in Wikipedia.

Using Knowledge from Wikipedia

Wikipedia has been intensively used recently to provide background knowledge for natural language processing. For example, to more accurately retrieve entities for a given query, the entity retrieval algorithm proposed in (Adafre, de Rijke, and Sang 2007) refers to the list and category information in Wikipedia. The entity ranking algorithm in (Vercoustre, Thom, and Pehcevski 2008) measures the importance of an entity using the link and category information in Wikipedia. Grineva, Grinev, and Lizorkin (2009) suggested a graph-based keyword extraction method using Wikipedia to weigh word importance in the document as well as to estimate the semantic relativeness between words. Mihalcea and Csomai (2007) defined a keyphraseness feature to link a document to its related knowledge facts in Wikipedia. People have also introduced various methods to measure word semantic relativeness based on the structure of Wikipedia, using in particular Wikipedia's category graph, such as (Zesch and Gurevych 2007), or using Wikipedia's inlink and outlink structures, e.g. (Milne and Witten 2008).

The keyword extraction and headline generation method introduced in this paper uses Wikipedia as the external knowledge repository for making keyword extraction and headline generation decisions. In our algorithm, we retrieve background knowledge of an input document through searching the Wikipedia XML corpus (Denoyer and Gallinari 2006). From the retrieved Wikipedia articles, we derive novel word features to reflect the background knowledge of the input document.

Also related to our work is a collection of recent studies on leveraging Wikipedia as a knowledge base for named entity disambiguation, e.g. (Bunescu and Pasca 2006; Cucerzan 2007; Han and Zhao 2009; Fader, Soderland, and Etzioni 2009).

Acquiring Document Background Knowledge using Wikipedia

Our Wikipedia based document background knowledge acquisition process consists of three main steps: 1) given a document, to generate a Wikipedia inquiry query for retrieving the document's background knowledge through searching the Wikipedia corpus; 2) execute the Wikipedia query to obtain a set of Wikipedia search result articles for the input document; and 3) derive from each search result article background knowledge relevant to the input document.

Generating a Wikipedia Search Query

To generate a query to find the most relevant background knowledge of a document via searching the Wikipedia corpus, we construct the query based on the key facts carried in the input document. This is done through selecting important content words from the input document. Specifically, we first apply a modified version of the TextRank algorithm (Mihalcea 2004) to detect important sentences in the input document. In the original TextRank algorithm, pairwise sentence similarity is based on the word matching degree of two sentences. We modified the way the pairwise sentence similarity is calculated. Instead of relying on word spelling matching as is done in (Mihalcea 2004), we measure sentence similarity through word semantic relativeness analysis (Pedersen, Patwardhan, and Michelizzi 2004). This modification helps us to more reliably detect sentence semantic similarity than the original TextRank method because pairwise word similarity is more accurately measured using semantics of words than through mere counting of the number of overlapping characters in the spellings of two words.

Given a few key sentences selected from the input document through the above process, we then perform stop word removal and word stemming over all the words in these key sentences. The remaining words constitute our Wikipedia search query.

Searching the Wikipedia Corpus

Once the Wikipedia search query for the input document is generated, we call on the full text search engine, Zettair (Billerbeck et al. 2004), to retrieve articles from the Wikipedia XML corpus that are related to the input document's key contents. The search results are returned as a ranked list of Wikipedia articles and their corresponding relativeness to the search query (query relativeness scores). We denote the set of retrieved Wikipedia articles as Π. The r-th Wikipedia article in the search result article set Π is denoted as p_r. The query relativeness score of article p_r is denoted as $z(p_r)$. We also denote the size of Π as N, i.e., the number of articles contained in the search result article set Π. Next we will explain how to extract the most essential background information from the Wikipedia search result article set Π.

Extracting Document Background Knowledge from Wikipedia Search Result Article Set

For each Wikipedia article retrieved in the previous step, we extract the following three types of background knowledge

of the input document: the link, category, and infobox information in a Wikipedia article.

1) Extracting Inlink Title Set and Outlink Title Set for a Wikipedia Search Result Article Wikipedia is organized as a hyperlinked text corpus, which allows readers to browse and navigate through its content following the link structure. An inlink points from another Wikipedia article to the current Wikipedia article whereas an outlink points from the current Wikipedia article to another Wikipedia article. Both inlink and outlink provide additional related information to help the readers better understand the topic(s) discussed in the current Wikipedia article.

To extract the inlink and outlink information from an article in our Wikipedia search result set Π, we first extract all the hyperlinks embedded in a Wikipedia article. We discard two types of hyperlinks: external links and internal links. External links point from a Wikipedia article to a webpage on the Internet outside Wikipedia. We discard such hyperlinks because we only intend to use knowledge extracted from Wikipedia to help our keyword extraction process. We do not want to utilize knowledge from the broad Internet because the quality of knowledge there cannot be guaranteed. Internal links are references from a certain document position in a Wikipedia article to another position in the same Wikipedia article. Since they do not link to a new Wikipedia article, they do not provide additional information.

After discarding the above two types of hyperlinks, the remaining hyperlinks are all the outlinks of the article. To extract the inlinks of a Wikipedia article, we use the MediaWiki API (MediaWiki 2009). For each article p_r in the Wikipedia search result set Π, we derive its inlink set, $IL(p_r)$, and outlink set, $OL(p_r)$, following the above procedure. For either set $IL(p_r)$ or $OL(p_r)$, we extract all the titles of the articles contained in the set, which respectively gives us an *inlink title set*, $IT(p_r)$, and an *outlink title set*, $OT(p_r)$, for the Wikipedia search result article p_r.

2) Extracting Category Set for a Wikipedia Search Result Article Category is another important type of information in Wikipedia, which appears at the bottom of a Wikipedia article to indicate the key topics covered in the article. Category information is organized as a graph structure in Wikipedia. Users can navigate through the graph to locate Wikipedia articles of their interests. Each Wikipedia article may be associated with a number of categories. We keep track of the categories that a Wikipedia article p_r is associated with in the *category set* of the article, which is denoted as $C(p_r)$.

3) Extracting Infobox Attribute Value Set for a Wikipedia Search Result Article In Wikipedia, infobox is generated using a certain infobox template. An infobox template typically carries several attributes for describing the key facts of the subject in a Wikipedia article. To make a template widely useable, editors often choose some common words as attribute names. Hence attribute names themselves carry very little entity specific information; but the most revealing texts would be used as attribute values. In view of this fact, we extract all the infobox attribute values of a Wikipedia article p_r, and organize them into an *infobox attribute value set* $IV(p_r)$.

Novel Word Features for Keyword Extraction and Headline Generation

In the following, we will first introduce some novel word features for keyword extraction and headline generation. We will then discuss how to extract keywords from a document through a learning based approach using these new word features. After that, we will look at how to generate a document's headline based on its keyword extraction result.

Novel Word Features

1) Word Inlink and Outlink Features For every word x_i in an input document, we derive a word inlink feature, $S_I(x_i)$, and an outlink feature, $S_O(x_i)$, using the inlink and outlink information in the input document's corresponding Wikipedia search result article set Π, as follows:

$$S_I(x_i) \triangleq \frac{\sum_{p_r \in \Pi} \left[z(p_r) \cdot \sum_{k \in IT(p_r)} \sigma_1(x_i, k) \right]}{\sum_{p_r \in \Pi} z(p_r) \cdot |IT(p_r)|}; \quad (1)$$

$$S_O(x_i) \triangleq \frac{\sum_{p_r \in \Pi} \left[z(p_r) \cdot \sum_{k \in OT(p_r)} \sigma_1(x_i, k) \right]}{\sum_{p_r \in \Pi} z(p_r) \cdot |OT(p_r)|}, \quad (2)$$

In the above, $z(p_r)$ is the query relativeness score of the Wikipedia article p_r. $IT(p_r)$ and $OT(p_r)$ are respectively the inlink and outlink title set of the Wikipedia article p_r. $|X|$ is the size of the set X. $\sigma_1(x_i, k)$ is the pairwise word semantic similarity (Pedersen, Patwardhan, and Michelizzi 2004). k is a word either in the inlink title set $IT(p_r)$ or in the outlink title set $OT(p_r)$. By the above definition, the more semantically similar a word x_i is to words in the inlink title set $IT(p_r)$ or words in the outlink title set $OT(p_r)$, the larger would be x_i's inlink feature value, $S_I(x_i)$, or outlink feature value, $S_O(x_i)$.

2) Word Category Feature Similarly, we introduce a word category feature S_C using the category information of every article in the input document's Wikipedia search result article set Π. Our calculation method is very similar to the way we compute the word inlink and outlink features in the above. The main difference is that we use the word similarity σ_1 based on the WordNet graph when deriving the word inlink and outlink features while here we use the word category similarity (Zesch and Gurevych 2007) which is based on the Wikipedia category graph to compute the word category feature.

3) Word Infobox Feature We also use Wikipedia's infobox information to derive a word infobox feature S_F. The definition over S_F is very similar to the inlink and outlink features of a word with the only difference being that we replace the inlink title set $IT(p_r)$ with the infobox value set $IV(p_r)$.

4) Word–Document Genre Fitness Feature Stewart (2008) recently proposed document genre based features for document summarization. In this paper, we argue that the genre of a document also has a major impact on which word shall be extracted as a keyword or adopted as a headline word for an input document.

5) Common Word Features Directly Derivable from the Input Document In our method, we also use the following common word features which can be directly derived from the input document: word frequency feature, word position features, specific name feature, relative word length feature, and conclusion sentence feature.

Keyword Extraction through a Learning based Approach

Once all the word features introduced in the above are derived, we can then apply a machine learning based approach to extract the keywords. We treat the document keyword extraction problem as a classification problem, in which a word in a document is classified as either a keyword or not a keyword. The training set for our learning based approach consists of full text articles and their corresponding headlines. For simplicity, all the non-stop words in a document's headline are considered as the keywords of the document. In our current experiment, we downloaded 817 articles from http://news.google.com/ to establish the training set, where the word lengths for the majority of articles are between 300 to 400 words. Other more sophisticated keyword corpus construction methods can also be employed, which however is not the focus of our work here. One clear advantage of our above automatic keyword-document set preparation method is that we can obtain a very large set of training examples without any human labeling efforts. We apply the support vector machine (SVM) method to the keyword extraction task.

Headline Generation based on Keyword Extraction

Using the keywords extracted from an input document, we can generate the document's headline. Noticing that simply putting together the keywords of a document as the document's headline would produce a piece of text with poor readability, we therefore employ the keyword clustering based headline generation procedure proposed in (Zhou and Hovy 2003) to construct a document's headline from the extracted keywords.

Experimentation

Experiments on Keyword Extraction

As there is no commonly available data set for keyword extraction, we first construct our own keyword extraction groundtruth data set through collecting 200 recent online articles posted on the BBC and CNN websites using a web crawler. We then asked 10 master's students in our computer science department to extract keywords manually from these articles. Every student was asked to extract 5 to 10 keywords from every article assigned to them. Each article was analyzed by four students. And we treat a manually identified keyword as the article's keyword if at least two students selected the word as the article's keyword. After carrying out this manual keyword extraction process, we constructed a data set consisting of 200 articles and their corresponding keywords.

Table 1: Performance comparison between different keyword extraction methods.

Keyword Extraction Method	Precision	Recall	F-rate
TF $\times$ IDF	0.210	0.312	0.251
Yahoo! Term Extraction	0.231	0.362	0.282
Wikify!	0.285	0.421	0.340
Community detection	0.312	0.435	0.373
Our method	**0.456**	**0.513**	**0.483**

Table 2: Comparison of different query generation methods on our algorithm's overall keyword extraction performance.

Query Generation Method	Precision	Recall	F-rate
(R.Soricut and D.Marcu 2007)	0.331	0.398	0.361
(Berger and Lafferty 1999)	0.397	0.452	0.423
Modified TextRank	**0.456**	**0.513**	**0.483**

We employ widely used precision, recall and F-rate measurements to evaluate the performance of our approach for keyword extraction. We compare the keywords of a document identified by our algorithm to the keywords of the document as labeled in the groundtruth dataset. We also implemented several existing keyword extraction methods including TF$\times$IDF (Salton and Buckley 1987), Yaoo! Term Extraction (Yahoo! 2010), Wikify! (Mihalcea and Csomai 2007), and community detection based keyword extraction algorithm (Grineva, Grinev, and Lizorkin 2009). We compare the performance of these peer methods and our algorithm for keyword extraction and report the results in Table 1. These results confirm the advantage of our method for keyword extraction.

We also explored different query generation methods to optimize the overall performance of our method for keyword extraction. For this purpose, we implemented three different query generation methods and experimentally compared the overall keyword extraction performance when employing each query generation method in the first step of our algorithm respectively. According to the results reported in Table 2, we can see that the modified TextRank method, currently employed in our algorithm, allows our keyword extraction approach to perform the best.

Experiments on Headline Generation

Table 3 gives some sample headline generation results along with counterparts produced by human editors. The underlying machine learning method employed for our keyword extraction process is the version of the SVM algorithm proposed by Yutaka et al. (2002). Even though the two sets of headlines are not exactly the same in terms of their wordings, we find that for all the five cases examined here, the headlines generated by our method successfully convey the

Table 3: Some example headlines generated using our approach and their counterparts authored by human editors.

Human editor authored headlines	Headlines generated using our approach
China urges flood-control workers to persist	flooding on the Yangtze and Songhua rivers evolved China today urged soldiers
Charles M. Schulz, creator of beloved Peanuts, dies at 77	Charles Schulz the creator of Peanuts was colon cancer
U.S. Republican Party holds first major presidential event	The Republican Party of the United States the party's presidential nominee
Food poisoning outbreak claims 11th victims in Scotland	The food poison outbreak in Lanarkshire Scotland the crisis
U.S. Korean warships stop Somali pirate attack	South Korean and American warships suspected pirates the U.S. Navy

Table 4: Performance comparison on different headline generation methods.

Headline Generation Method	ROUGE-1	ROUGE-2
Lead10 (R.Soricut and D.Marcu 2007)	0.208	0.111
Hedge Trimmer (Dorr, Zajic, and Schwartz 2003)	0.181	0.099
Topiary (Zajic, Dorr, and Schwartz 2004)	0.262	0.125
Template Filter (Zhou and Hovy 2004)	0.169	0.042
ISI (Jin and Hauptmann 2001)	0.141	0.075
WIDL (R.Soricut and D.Marcu 2007)	0.255	0.129
Our approach using SVM	**0.543**	**0.185**

key messages of the corresponding articles.

We also tested the efficacy of our novel word features for headline generation by comparing the performance of our headline generation procedure using our novel word features with the performance of other peer methods. Unlike the keyword extraction experiments for which there is no publicly available groundtruth dataset for evaluating the quality of a keyword extraction algorithm, there does exist some widely available, standardized dataset for evaluating the performance of headline generation algorithms. We have utilized the document corpus released by the Document Understanding Conference (DUC-2003) in this regard. In Table 4, we compare the performance of our algorithm with that of six other headline generation methods. The Lead10 method (R.Soricut and D.Marcu 2007) is a simple algorithm that extracts the first 10 words of the lead sentence of a document as the document's headline. The Hedge trimmer algorithm (Dorr, Zajic, and Schwartz 2003) uses linguistically motivated heuristics to guide the headline generation process. The Topiary's algorithm (Zajic, Dorr, and Schwartz 2004) combines linguistically motivated sentence compression technique with statistically selected topic terms to generate a headline. The template filter based algorithm (Zhou and Hovy 2004) utilizes a template based method to generate a headline. The ISI method (Jin and Hauptmann 2001) first generates a headline text using a sentence position model and then refines the generated headline in terms of its readability through a post-processing step. The WIDL algorithm (R.Soricut and D.Marcu 2007) performs the headline generation using statistical knowledge encapsulated in both WIDL-expressions and some language models. From the results of this set of comparison experiments, we can clearly see the superiority of our novel word features for headline generation, which can help produce headlines most similar to the ones authored by human editors.

Table 5: Performance of our method for headline generation using different subsets of our novel word features.

Features	ROUGE-1	ROUGE-2
All features	**0.543**	**0.185**
No inlink and outlink features	0.382	0.105
No category feature	0.463	0.146
No genre feature	0.412	0.132
No infobox feature	0.461	0.141
Common word features only	0.283	0.085

To quantitatively study the effectiveness of our novel word features for headline generation, we also conducted a series of controlled experiments where different subsets of features are used in our method for headline generation. We report the performance of these variants of our method in Table 5, the results of which clearly show the necessity and effectiveness of engaging all the novel word features introduced in this paper for headline generation.

Conclusion and Future Work

In this paper, we propose some novel word features for keyword extraction and headline generation. These new word features are derived through background knowledge of a document. The background knowledge is acquired via first querying Wikipedia, and then exploring the inlink, outlink, category, and infobox information of the Wikipedia search result article set. We also introduce a word-document genre fitness feature to observe the word selection bias imposed by the genre of a document. Experimental results have proved that using these novel word features, we can achieve superior performance in keyword extraction and headline generation to other state-of-the-art approaches.

Our current work uses Wikipedia as the source to acquire the background knowledge of a document. This is carried out on the basis of single Wikipedia articles. The hierarchical structure of Wikipedia is largely unutilized. In the future, we plan to explore the hierarchical structure of Wikipedia to derive more semantically revealing word features to assist keyword extraction and headline generation.

References

Adafre, S. F.; de Rijke, M.; and Sang, E. T. K. 2007. Entity retrieval. In *Proceeding of Recent Advances in Natural Language Processing (RANLP-07)*.

Berger, A., and Lafferty, J. 1999. Information retrieval as statistical translation. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 222–229.

Billerbeck, B.; Cannane, A.; Chattaraj, A.; Lester, N.; Webber, W.; E.Williams, H.; Yiannis, J.; and Zobel, J. 2004. RMIT university at TREC 2004. In *Proceedings of Text Retrieval Conference (TREC-04)*.

Bunescu, R. C., and Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, 9–16. Trento, Italy: The Association for Computer Linguistics.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-07)*, 708–716. Prague, Czech Republic: Association for Computational Linguistics.

Denoyer, L., and Gallinari, P. 2006. The Wikipedia XML corpus. *SIGIR Forum* 40(1):64–69.

Dewdney, N.; VanEss-Dykema, C.; and MacMillan, R. 2001. The form is the substance: classification of genres in text. In *Proceedings of the Workshop on Human Language Technology and Knowledge Management*, 1–8.

Dorr, B.; Zajic, D.; and Schwartz, R. 2003. Hedge trimmer: a parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Summarization*, 1–8.

Ercan, G., and Cicekli, I. 2007. Using lexical chains for keyword extraction. *Information Processing and Management* 43(6):1705–1714.

Fader, A.; Soderland, S.; and Etzioni, O. 2009. Scaling wikipedia-based named entity disambiguation to arbitrary web text. In *Proceedings of the Wiki-AI Workshop at IJCAI (WIKIAI-09)*.

Grineva, M.; Grinev, M.; and Lizorkin, D. 2009. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th International Conference on World Wide Web*, 661–670.

Han, X., and Zhao, J. 2009. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *CIKM '09: Proceeding of the 18th ACM Conference on Information and Knowledge Management*, 215–224. New York, NY, USA: ACM.

Jin, R., and Hauptmann, A. G. 2001. Title generation using a training corpus. In *Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Text Processing*, 208–215.

MediaWiki. 2009. API – MediaWiki, the free Wiki engine. http://www.mediawiki.org/w/index.php?title=API&oldid=254363.

Mihalcea, R., and Csomai, A. 2007. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 233–242.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, 233–242.

Mihalcea, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Interactive Poster and Demonstration Sessions of ACL 2004*.

Milne, D., and Witten, I. H. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of Wikipedia and AI Workshop at the AAAI-08 Conference (WikiAI-08)*.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. Wordnet::similarity – measuring the relatedness of concepts. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, 1024–1025.

R.Soricut, and D.Marcu. 2007. Abstractive headline generation using WIDL-expressions. *Information Processing and Management* 43(6):1536–1548.

Salton, G., and Buckley, C. 1987. Term-weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA.

Stewart, J. G. 2008. *Genre Oriented Summarization*. Ph.D. Dissertation, Cargegie Mellon University.

Vercoustre, A.-M.; Thom, J. A.; and Pehcevski, J. 2008. Entity ranking in Wikipedia. In *Proceedings of ACM Symposium on Applied Computing*, 1101–1106.

Yahoo! 2010. Yahoo! term extraction API, http://developer.yahoo.com/search/content/v1/termextraction.html.

Yutaka, T. H.; Sasaki, Y.; Isozaki, H.; and Maeda, E. 2002. Ntt's text summarization system for DUC-2002. In *Proceedings of the Document Understanding Conference*, 104–107.

Zajic, D.; Dorr, B.; and Schwartz, R. 2004. BBN/UMD at DUC-2004: Topiary. In *Proceedings of the 2004 Document Understanding Conference (DUC-04) at NLT/NAACL 2004*, 112–119.

Zesch, T., and Gurevych, I. 2007. Analysis of the Wikipedia category graph for NLP applications. In *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT-07)*.

Zhang, K.; Xu, H.; Tang, J.; and Li, J.-Z. 2006. Keyword extraction using support vector machine. In *Lecture Notes in Computer Science: Advances in Web-Age Information Management*, 85–96.

Zhou, L., and Hovy, E. 2003. Headline summarization at ISI. In *Proceedings of the HLT/NAACL Workshop on Automatic Summarization/Document Understanding Conference (DUC-03)*.

Zhou, L., and Hovy, E. 2004. Template-filtered headline summarization. In *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out*, 56–60.