Improving Keyphrase Extraction from Web News by Exploiting Comments Information

Zhunchen Luo, Jintao Tang and Ting Wang

College of Computer, National University of Defense Technology,
410073 Changsha, Hunan, China
{zhunchenluo,tangjintao,tingwang}@nudt.edu.cn

Abstract. Automatic keyphrase extraction from web news is a fundamental task for news documents retrieval, summarization, topic detection and tracking, etc. Most existing work generally treats each web news as an isolated document. With the rapidly increasing popularity of Web 2.0 technologies, many web news sites provide various social tools for people to post comments. These comments are highly related to the web news and can be considered as valuable background information which can potentially help improve keyphrase extraction. In this paper we propose a novel method to integrate the comment posts into the task of extracting keyphrases from a web news document. Since comments are typically more casual, conversational, and full of jargon, we introduce several strategies to select useful comments for improving this task. The experimental results show that using comments information can significantly improve keyphrase extraction from web news, especially our comments selection method, using machine learning technology, yields the best result.

Keywords: Keyphrase Extraction, Comments, Web News, Machine Learning

1 Introduction

As thousands of web news are posted on the Internet everyday, it is a challenge to retrieve, summarize, classify or mine this enormous repository effectively. Keyphrases can be seen as condensed represents of web news documents which can be used to help these applications. However only a small minorities of documents have author-assigned keyphrases and manually assigning keyphrases for each document is very laborious. Therefore it is highly desirable to extract keyphrases automatically. In this paper we consider this task in web news scenario.

Most existing work on keyphrase extraction from a web news document only considers the internal information, such as the phrase's Tf-idf, position and other syntactic information, which neglects the external information of the document. Some researches also utilize background information. For example, Wan *and* Xiao proposed a method, which selected similar documents for a given document from a large documents corpus, to improve the performance of keyphrase extraction [14]. Mihalcea *and* Csomai used Wikipedia to provide background knowledge for this task [10]. However both of these work needs to retrieve topic-related documents as external information, which is

a non-trivial problem. Fortunately, nowadays many web news sites provide various social tools for community interaction, including the possibility to comment on the web news. These comments are highly related to the web news documents and naturally bound with them. The motivation of our work is that comments are valuable resource providing background information for each web news document, especially some useful comments contain rich information which is the focus of the readers. Hence, in this paper we consider exploiting comments information to improve keyphrase extraction from the web news documents.

Unfortunately, although the main entry and its comments are certainly related and at least partially address similar topics, they are markedly different in several ways [16]. First of all, their vocabulary is noticeably different. Comments are more casual, conversational, and full of jargon. They are less carefully edited and therefore contain more misspellings and typographical errors. There is more diversity among comments than within the single-author post, both in style of writing and in what commenters like to talk about. Simply using all the comments as external information does not improve the performance of keyphrase extraction (see section 5.2). Therefore, we propose several methods to select useful comments, based on some criterions such as textual similarity and helpfulness of the comment. The experimental results show our comment selection approaches can improve keyphrase extraction for web news. Especially our machine leaning approach which integrates many factors of comments can significantly improve this task.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes the framework of our approach to extract keyphrases from a web news document. In section 4 we propose the strategies for selecting useful comments. Section 5 presents and discusses the evaluation results. Lastly we conclude the paper.

2 Related Work

Most existing work only uses the internal information of a document for keyphrase extraction. The simplest approach is to use a frequency criterion (or Tf-idf model [12]) to select keyphrases in a document. This method was generally found to give poor performance [3]. However, Hasan recently conducted a systematic evaluation and analysed some algorithms on a variety of standard evaluation datasets, his results indicated that Tf-idf remained offering very robust performance across different datasets [4]. Another important clue for keyphrase extraction is the location of phrase in the document. Kea [2] and GenEx [13] used this clue for their studies. Hulth added more linguistic knowledge, such as syntactic features, to enrich term representation, which were effective for keyphrase extraction [5]. Mihalcea *and* Tarau firstly proposed a graph-based ranking method for keyprase extraction [11] and currently graph-based ranking methods have become the most widely used approaches [3, 8, 9]. However, all the approaches above never consider the external information when extracting keyphrases.

Another important information, which can be used for keyphrase extraction, is external resource. Wikipedia[1] has been intensively used to provide background knowledge. Mihalcea *and* Csomai used the link structure of Wikipedia to derive a novel word

[1] `http://www.wikipedia.org/`

feature for keyphrase extraction [10]. Grineva *et al.* utilized article titles and the link structure of Wikipedia to construct a semantic graph to extract keyphrases [3]. Xu *et al.* extracted the in-link, out-link, category, and inbox information of the documents related the article in Wikipedia for this task [15]. All approaches using Wikipedia significantly improve the performance of keyphrase extraction. Another external resource is neighbor documents. Wan *and* Xiao proposed an idea that a user would better understand a topic expressed in a document if the user reads more documents about the same topic [14]. He used a textual similarity measure (e.g., cosine similarity) to find documents topically closed to the document from a large documents corpus. These neighbor documents are beneficial to evaluate and extract keyphrases from the document. However using the external resource introduced above needs to retrieve topic-related documents, which is a non-trivial problem. Unlike these resources, we exploit comments of web news as external information. This information is topic-related to the original document. Moreover, the characteristic of naturally bound with the original news makes the comments easier to obtain.

3 Framework

A web news document has two parts: an article and comments. The article contains the text of event written by the author. Comments can be seen as the discussion of the article including the attitude of readers. Most of comments are topic-related to the article. Especially some comments contain rich information which is focused by the readers. We call the comments, which provide additional information for readers to better understand the content of the article, **useful comments**. Intuitively, these useful comments are expected to be beneficial to evaluate and extract the salient keyphrases from the article. However traditional keyphrase extraction methods only consider the information of the article and ignore its comments information. This study we propose an approach exploiting comments information to improve keyphrase extraction for the web news documents.

Figure 1 gives the framework of our approach for keyphrase extraction. First we segment a web news document into the article part and the comments part based on its natural structure. Next, since not all comments are helpful for keyphrase extraction (see section 5.2), we use a comments selector, based on some strategies, to select useful comments from the comments part. Then we integrate the selected comments with the article part to form a new document. At last we use a keyphrase extractor to extract phrases from the new document. The extracted phrases, occurred in the article, are taken as keyphrases. Our approach can be easily incorporated with any state-of-the-art keyphrase extractor to extract keyphrases from web news.

Since some comments contain noise information such as advertisement, irrelevant texts and texts only containing meaningless word emoticons. For example, a comment *"Who cares!"* just reflects a reader's attitude to the news without any useful information for other readers to understand the article. Hence, the key point of our approach is to find the useful comments for keyphrase extraction. We propose three strategies to select useful comments, which are expected to find the article's keyphrases with higher accuracy. The first is selecting comments which is similar to the original article, called

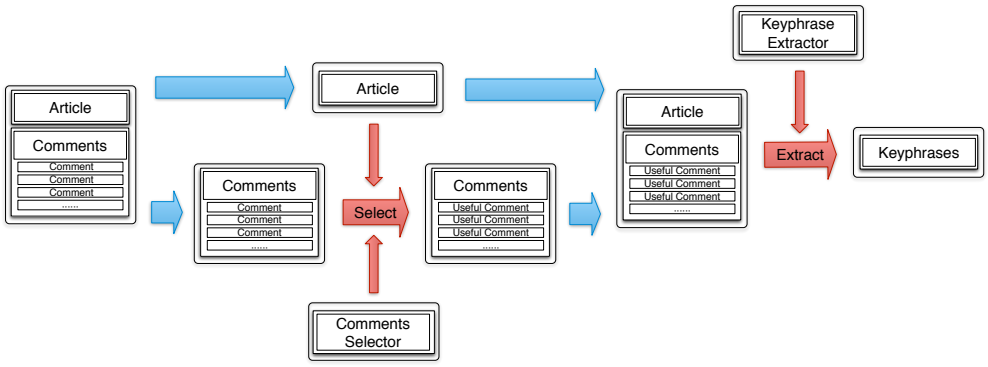

Fig. 1. Framework of Keyphrase Extraction from a Web News Document

Similar Comments Selector. The idea of this approach is the same as Wan *and* Xiao [14]. Many web news sites provide ratings setting about the comments for other readers (e.g, thumb up or thumb down votes). These meta ratings information can help filter useful comments more efficiently. We call the approach based on ratings information **Helpfulness Comments Selector**. Furthermore, since existing technologies of natural language processing and machine learning provide strong supports to mine the potential knowledge, we use these technologies to select useful comments, called **KeyInfo Comments Selector**. Next section we will introduce the three comments selectors in detail.

4 Selecting Useful Comments

4.1 The Similar Comments Selector

Wan *and* Xiao proposed a method to select similar documents for a given document to improve the performance of keyphrase extraction [14]. Their assumption is that multiple documents within an appropriate cluster context usually have mutual influences and contain useful clues, which can help to extract keyphrases from each other. For example, two documents about the same topic “earthquake” would share a few common phrases, e.g. “earthquake”, “victim”, and they can provide additional knowledge for each other to better evaluate and extract salient keyphrases from each other.

The similar comments selector adopts the same idea to find useful comments. Firstly it uses the widely used cosine measure to evaluate document similarity between the web news article and its comments. The term weight is computed by Tf-idf. Then the selector chooses top N comments with the highest similarity values as the useful comments for the web news keyphrase extraction.

4.2 The Helpfulness Comments Selector

Due to the lack of editorial and quality control, comments on web news vary greatly. The low-quality comments may hurt the performance of keyphrase extraction. Many web news sites also allow readers to vote for the helpfulness of each comment and provide a function to assess the quality. The helpfulness function h is usually defined as follow:

$$h = \frac{Num_{thumbup}}{Num_{thumbup} + Num_{thumbdown}} \tag{1}$$

where $Num_{thumbup}$ is the number of people, who consider the comment is helpful. $Num_{thumbdown}$ is the number of people, who consider the comment is unhelpful. This function actually provides a direct and convenient way to estimate the utility value of a given comment. Therefore, we use the helpfulness of comments based on this function to select useful comments. The top N comments with highest scores voted by at least 5 users are selected for the following keyphrase extraction step.

4.3 The KeyInfo Comments Selector

A simple idea is that given an article, if the comments include more keyphrases, they are more likely to be useful comments than the other comments which have less or no keyphrase, since these comments contain more information which the readers focus on. Intuitively, integrating these useful comments into original article is expected to improve the performance of keyphrase extraction. However, we can not obtain the accurate number of keyphrases in each comment, since it is the final purpose of the task. Fortunately, some factors of comments are related to the probability of comments containing more keyphrases such as the textual similarity and helpfulness of comments. Moreover, existing natural language processing technologies can help us find many factors related to the probability of comments containing keyphrases, and using machine learning technologies, based on these factors, can also estimate the probability accurately. Therefore, we propose another comments selector called KeyInfo comments selector using natural language processing and machine learning technologies.

The principle of the KeyInfo comments selector is that if a shorter comment contains more keyphrases, it is more likely to improve the performance of keyphrases extraction from the web news when integrating it into the original article. We define a function k called keyInfo function as follow:

$$k = \frac{Num_{keyphrase}}{Num_{word}} \tag{2}$$

where $Num_{keyphrase}$ is the number of keyphrases in the comment and Num_{word} is the number of words in the comment. We choose the top N comments with highest score of k as the useful comments for keyphrase extraction.

The key point of this approach is to predict the value of $Num_{keyphrase}$. We take this prediction as a regression-learning problem. Firstly, some web news and their comments are collected as training data. These news documents have been originally

annotated keyphrases. Then using these gold standard keyphrases, each comment's $Num_{keyphrase}$ can be easily obtained. Some features, which can potentially help to predict the $Num_{keyphrase}$, are designed for each comment. Next each comment is represented by a feature vector. Then a classical regression tool-simple linear regression (SLR) implemented in WEKA[2] is used to train a prediction model, called KeyInfo model. Lastly, give a new comment represented as a feature vector, this model can predict its $Num_{keyphrase}$ value.

To the features for each comment, textual similarity and helpfulness are not enough. Since the most similar comments are similar to the original entire news article text, the probability of containing keyphrases in each part of the article text, however, is different. For example, the headline of article, the beginning and the end part of article may have higher probability than another parts of text. If a comment is more similar to these key parts of the article, it is more likely to be useful comment for keyphrase extraction. Additionally, for the comments with high score of helpfulness, the phenomenon of "rich-get-richer" might hurt the performace of this approach. The reason is that, how fast a comment accumulate votes depends on the number of votes they already have [7] and this led to a bias to the comments with more user voting. Therefore, we develop more features, based on natural language processing technologies, for the KeyInfo model. These features are designed to maintain the advantages and overcome the disadvantages of two approaches introduced above. Table 1 shows all the features for the KeyInfo Model. All developed features are classified into five categories: **Structural**, **Similar**, **Lexical**, **Syntactic**, **Semantic**, and **Meta-data**. We use *Stanford Log-linear Part-Of-Speech Tagger*[3] to tag the comment and use *General Inquirer*[4] for sentiment analysis.

5 Experiment

We empirically evaluate our approach for keyphrase extraction from a web news document, especially comparing the performance of various strategies to select useful comments for this task. In addition we conduct in-depth analysis of the differences among useful comments sets chosen by different comments selectors.

5.1 Experimental Setup

The AOL Corpus We construct our own keyphrase extraction data set by collecting 60 web news posted on the AOL websites[5]. All these news are world news posted from November 16th 2010 to November 26th 2010. Every news document contains at least 20 comments. The main reason we choose AOL web news for the experiment is that every web news document has its own tags. We take the tagged phrases that occur in the article as gold standard keyphrases. Table 2 provides an overview of the AOL corpus.

[2] http://www.cs.waikato.ac.nz/ml/weka/
[3] http://nlp.stanford.edu/software/tagger.shtml
[4] http://www.wjh.harvard.edu/~inquirer/
[5] http://www.aol.com/

Table 1. Features of the KeyInfo Model

Feature category	Feature name	Description
Structural	NumTokens	The number of tokens in the comment
	NumSentences	The number of sentences in the comment
	NumAskMarks	The number of ask marks in the comment
	NumExclamationMarks	The number of exclamation marks in the comment
Similar	ArticleSimilar	The similarity between the comment and the article
	TitleSimilar	The similarity between the comment and the title
	HelpfulnessCommentsSimilar	The similarity between the comment and other comments with the highest score of helpfulness (the number of other comments is 5)
Syntactic	RatioNouns	The percentage of nouns in the comment
	RatioVerb	The percentage of verb in the comment
	RatioAdj	The percentage of adjective in the comment
	RatioAdv	The percentage of adverb in the comment
Semantic	NumPosWord	The number of positive words in the comment
	NumNegWord	The number of negative words in the comment
MetaData	NumThumbUps	The number of thumb up votes in the comment
	NumThumbDowns	The number of thumb down votes in the comment
	HelpfulNess	The helpfulness score of the comment
	TimeLines	The number of comments having been posted before this comment

Table 2. AOL Corpus Statistics

	Ave	Max	Min
#Tokens in Article	570.77	1558	162
#Comments	103.98	316	22
#Gold Standard Keyphrases	7.6	17	3

Evaluation Metric For the evaluation of keyphrase extraction results, the automatic extracted keyphrases are compared with the gold standard keyphrases. The precision $p = count_{correct}/count_{extrator}$, recall $r = count_{correct}/count_{gold}$, and F1 score $f = 2pr/(p + r)$ are used as evaluation metrics, where $count_{correct}$ is the total number of correct keyphrases extracted by the extractor, $count_{extrator}$ is the total number of automatic extracted keyphrases, and $count_{gold}$ is the total number of gold keyphrases.

The Keyphrase Extractors Most state-of-the-art keyphrase extractors can be used in our approach. Hasan conducted a systematic evaluation and analysed many unsupervised keyphrase extraction methods on a variety of standard evaluation datasets. He found that Tf-idf and SingleRank extractors give very robust performance across different datasets [4]. Therefore, we use these two keyphrase extractors in our experiment. A publicly available implementation of these two extractors can be downloaded[6].

Tf-idf Keyphrase Extractor assigns a score to every term in a document based on its frequency (term frequency, tf) and the number of other documents containing this term (inverse document frequency, idf). Given a document the extractor first computes the Tf-idf score of each candidate word, then extracts all the longest n-grams consisting of candidate words and scores each n-gram by summing the Tf-idf scores of its constituent unigrams, and finally output the top N n-grams as keyphrases.

SingleRank Keyphrase Extractor is expanded from **TextRank Keyphrase Extractor** [11, 14]. Both of these two extractors take a text represented by a graph, in which each vertex corresponds to a word type and the edge connects two vertices if the two words co-occur within a window of W words in the associated text. The goal is to compute the score of each vertex, which reflects its importance. The two extractors use the word types that correspond to the highest-scored vertices to form keyphrases for the text. Some graph-based ranking algorithms such as Google's PageRank [1] and HITS [6], which compute the importance of vertices, can be easily integrated into the TextRank model and SingleRank model. There are some differences between these two extractors. First, each edge has a weight equal to the number of times the two corresponding word types co-occur within a window of W words in a SingleRank graph, while each edge has the same weight in a TextRank graph. Second, SingleRank only uses high score sequence nouns and adjectives words to form keyphrases, but any high score sequence words can form keyphrases in TextRank.

5.2 Evaluation of the Proposed Approaches

First we investigate all the comments information can help keyphrase extraction. We integrate all comments into their related articles in our AOL corpus to form a new dataset. We extract 20 keyphrases in each news document for evaluation using **Tf-idf Keyphrase Extractor** and **SingleRank Keyphrase Extractor** respectively. Table 3 shows the result. We can see that, compared the approaches extracted keyphrases from original article (**Base_Tf-idf** and **Base_SingleRank**), the approaches adding all comments into the article (**All_Tf-idf** and **All_SingleRank**) decrease the performance of extracting keyphrases. We conducted a paired t-test between the results of these methods and found statistically significant differences (at $p = 0.05$). All these means simply using all comments can not help for keyphrase extraction.

Next we investigate some useful comments can help keyphrase extraction. We ranked all the comments based on keyInfo function k. We use the gold keyphrases in each article to calculate the accurate $Num_{keyphrase}$ value for each comment. For each web news document, we choose the top 15 comments with high value of k and integrate them into the original article (called **Oracle_Tf-idf** and **Oracle_SingleRank**), which achieve the

[6] `http://www.utdallas.edu/~ksh071000/code.html`

Table 3. Performance of All Comments Testing

	Precision	Recall	F1
Base_Tf-idf	0.139	0.366	0.202
All_Tf-idf	0.153	0.259	0.192
Base_SingleRank	0.120	0.316	0.174
All_SingleRank	0.096	0.163	0.121

best result of keyphrase extraction in our experiment (see Table 4). Moreover, they are oracle tests and the results are significantly different (at $p = 0.05$). It means there is a subset of useful comments, which can help for keyphrase extraction, in comments part for each web news (see Figure 1).

Table 4. Performance of Useful Comments Testing

	Precision	Recall	F1
Base_Tf-idf	0.139	0.366	0.202
Oracle_Tf-idf	0.165	0.432	0.239
Base_SingleRank	0.120	0.316	0.174
Oracle_SingleRank	0.146	0.382	0.211

At last we investigate the performance of our three comments selectors, choosing top 15 high score comments, for keyphrase extraction. The **Similar_Tf-idf (SingleRank)** and **Helpfulness_Tf-idf (SingleRank)** methods used the measures of comments' similar and helpfulness score to choose the comments. To **KeyInfo_Tf-idf (SingleRank)** approach, we used 5 new AOL web news comments excluding AOL corpus to train a regression model, which can predict the $Num_{keyphrase}$ of each new comment. The 5 new documents contain 524 comments and each document has gold standard keyphrases. We extract all features in Table 1 to represent each comment as a feature vector for training. The trained model can be used to predict the KeyInfo score of a new comment. Table 5 gives the comparison results of various selecting methods for keyphrase extraction. We can see that the performance of most methods integrating comments outperform the **Base_Tf-idf(SingleRank)** for keyphrase extraction. It shows comments information indeed can help for keyphrase extraction. We conducted a paired t-test between the results and found all methods significantly improve the keyphrase extraction (at $p = 0.05$) except **Helpfulness_Tf-idf**. Especially the **KeyInfo_Tf-idf (SingleRank)** achieves the best result. It shows similar and helpfulness information are helpful for selecting useful comments. Moreover, using more knowledge, extracted based on natural language processing and machine learning technologies, can collect more useful comments. All these shows exploiting comments information are effective for keyphrase extraction.

We are interested in the detail of comments selected by these three selectors. We compare three selected comment sets with the comments selected based on the accurate $Num_{keyphrase}$ value (see Oracle Test). All of comparisons use the top 15 high score comments for each document. Table 6 gives the result. We can see that the KeyInfo

Table 5. Performance of Extracting 20 Keyphrases

	Precision	Recall	F1
Base_Tf-idf	0.139	0.366	0.202
Similar_Tf-idf	0.151	0.393	0.218
Helpfulness_Tf-idf	0.144	0.379	0.209
KeyInfo_Tf-idf	0.157	0.406	0.226
Base_SingleRank	0.120	0.316	0.174
Similar_SingleRank	0.134	0.324	0.190
Helpfulness_SingleRank	0.133	0.332	0.190
KeyInfo_SingleRank	0.144	0.331	0.201

selector collect the largest most useful comments. It demonstrates our machine learning approach is effective to find more comments with high function K value for keyphrase extraction.

Table 6. Useful Comments Sets Comparing Evaluation

	Number
$\text{Set}_{Oracle} \cap \text{Set}_{KeyInfo}$	437
$\text{Set}_{Oracle} \cap \text{Set}_{Helpfulness}$	331
$\text{Set}_{Oracle} \cap \text{Set}_{Similar}$	300

6 Conclusion

In this paper we propose a novel approach to use comments information for keyphrase extraction from web news documents. Since comments are full of noisy and low-quality data, simply integrating comment into an original article does not improve the performance of this task. Therefore we give three strategies to select useful comments. The experimental results show that our comments selection approaches can improve keyphrase extraction from web news. Especially our machine leaning approach which considers many factors of comments can significantly improve this task.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (Grant No. 61170156 and 61202337) and a CSC scholarship.

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the seventh international conference on World Wide Web 7. pp. 107–117. WWW7, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands (1998)

2. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. pp. 668–673. IJCAI '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
3. Grineva, M., Grinev, M., Lizorkin, D.: Extracting key terms from noisy and multitheme documents. In: Proceedings of the 18th international conference on World wide web. pp. 661–670. WWW '09, ACM, New York, NY, USA (2009)
4. Hasan, K.S., Ng, V.: Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 365–373. COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
5. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. pp. 216–223. EMNLP '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003)
6. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM 46(5), 604–632 (Sep 1999)
7. Liu, J., Cao, Y., Lin, C., Huang, Y., Zhou, M.: Low-quality product review detection in opinion summarization. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 334–342 (2007)
8. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 366–376. EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
9. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1. pp. 257–266. EMNLP '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
10. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. pp. 233–242. CIKM '07, ACM, New York, NY, USA (2007)
11. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP. pp. 404–411 (2004)
12. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24(5), 513–523 (Aug 1988)
13. Turney, P.D.: Learning to extract keyphrases from text. CoRR cs.LG/0212013 (2002)
14. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of the 23rd national conference on Artificial intelligence - Volume 2. pp. 855–860. AAAI'08, AAAI Press (2008)
15. Xu, S., Yang, S., Lau, F.C.M.: Keyword extraction and headline generation using novel word features. In: AAAI (2010)
16. Yano, T., Cohen, W., Smith, N.: Predicting response to political blog posts with topic models. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 477–485. Association for Computational Linguistics (2009)