An Automatic Online News Topic Keyphrase Extraction System

Canhui Wang, Min Zhang, Liyun Ru, Shaoping Ma
State Key Lab of Intelligent technology & systems,
Tsinghua National Laboratory for Information Science and Technology,
CS&T Department, Tsinghua University, Beijing, 100084, China P.R.
wangcanhui@gmail.com

Abstract

News Topics are related to a set of keywords or keyphrases[1]. Topic keyphrases briefly describe the key content of topics and help users decide whether to do further reading about them. Moreover, keyphrases of a news topic can be considered as a cluster of related terms, which provides term relationship information that can be integrated into information retrieval models. In this paper, an automatic online news topic keyphrase extraction system is proposed. News stories are organized into topics. Keyword candidates are firstly extracted from single news stories and filtered with topic information. Then a phrase identification process combines keywords into phrases using position information. Finally, the phrases are ranked and top ones are selected as topic keyphrases. Experiments performed on practical Web datasets show that the proposed system works effectively, with a performance of precision=70.61% and recall=67.94%.

1. Introduction

News stories are gathered from many Websites and organized into news topics by practical Web applications like *Google News*. Topics are constructed and updated online automatically, using the techniques of Topic Detection and Tracking (TDT) [1]. News TDT results have greatly facilitated users who want to know about "what's new" or "what's going on". However, with the vast amount of news topics created and updated all the time, it is almost impossible for users to view them all. It will be much convenient if all topics are represented with a few keyphrases respectively. Users can grasp the key content of a topic within several seconds by viewing the keyphrases. Afterwards they can make decisions about whether to do further reading about the topic.

Keyphrases of a document briefly describe the content of it. They are widely used in text indexing, summarization, and categorization [15]. An obvious application of topic keyphrases is topic summarization. News topics are usually described by 5W and 1H, i.e. "when, what, who, where, why and how". The keyphrases of news topics should be as much concerned to 5W1H as possible [17], in order to provide a brief description of topic content for users' quick glances.

Another application of topic keyphrases relates to information retrieval. Keyphrases are formed by combining keywords in adjacent occurrence positions. The keywords of a news topic can be considered as a cluster of related terms, which provides term relationship information and query context that can be integrated into information retrieval models [3, 4, 6]. For instance, a recent topic about "Jie Zheng, a Chinese tennis athlete, enters the final four in Wimbledon" has keywords like *Jie Zheng*, *China*, *final four*, *tennis*, *Wimbledon* and etc. So if users input a query "Jie Zheng, final four" to search engines on the day the topic took place, their intention was most probably to find out whether Jie Zheng had entered the final four in Wimbledon 2008. However, without *Wimbledon* in the query, it's possible to retrieve documents about "Jie Zheng failed to enter the final four in NASDAQ-100 in March 2006" and assign high ranks to them, especially if the search engine lacks a good mechanism for timely results. It's useful to discover the relationship among *Jie Zheng*, *final four* and *Wimbledon*, because *Wimbledon* can be used as query context for the query "Jie Zheng, final four". In [3, 4, 6], term relationships are extracted from a fixed document collection, and therefore recent relationships are not able to be found out. News topics contain the latest clusters of related terms, so keyphrases of news topics provide fresh term relationships for search engines to make their search results more relevant and timely.

Previous work mainly focuses on keyword extraction from a document, e.g. a news story, a journal article or a book [8, 10, 15, 17, 22, 23, 26, 27, 30]. A news topic is composed of news stories about the same people or things, so topic keyphrase extraction can make use of information from multi-stories, which generates keyphrases as representations for different aspects of a topic.

The problem to be investigated is: how to extract keyphrases from news topics online automatically? Motivated by the problem, we propose a news topic

[1] In this paper, "keyphrase" has the same meaning as "keyword" in most circumstances except that the former is more complete than the latter in syntactic and semantic structures.

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.225

keyphrase extraction system in this paper. The state-of-the-art TDT techniques are used to organize news pages from a lot of news Websites into topics. An aging theory is added in the TDT process. Unlike previous methods, our system firstly extracts keyword candidates from single news stories, filters them with topic information and then combines them into phrase candidates using position information. Finally, the phrases are ranked and top ones are selected as topic keyphrases.

The rest of the paper is organized as follows: Section 2 gives a brief review of related work. Section 3 describes the topic detection and tracking algorithm based on burstiness of terms and aging theory. The topic keyphrase extraction algorithm is proposed in Section 4. We describe the experimental data and results in Section 5, followed by the conclusion and a discussion of future work in Section 6.

2. Related work

Our system mainly relates to previous work on topic detection and tracking (TDT) and keyword extraction.

TDT are intended to structure news stories from newswires and broadcasts into topics [1]. Approaches in TDT were mainly variants and improvements of the single pass method and agglomerative clustering algorithms [2, 5, 9, 16, 18, 19, 21, 24, 28, 29]. Although [5] concluded that time information "did not help" improve the new event detection results, some recent work has utilized the aging theory and achieved good performance in TDT [7]. The state-of-the-art TDT techniques and aging theory are used to generate topics from news stories in our system.

The relationship between topics and corresponding keywords has been studied and utilized previously, mostly in the retrospective way: topic hierarchy construction based on identification of bursty periods of features [11]; using "the time information to determine a set of bursty features which may occur in different time windows" and detecting bursty topics by grouping bursty features [12]; topic detection based on identification of both aperiodic and periodic features' bursts [13]; finding top bursty topics by identifying bursty words [14]; and so on. Previous approaches listed above analyzed the characteristics of features from a fixed corpus on the whole timeline, and hence have to be adjusted to suit to online use. Our system deals with dynamic-increasing news data online, and makes use of an aging theory in topic detection and tracking.

There are supervised and unsupervised algorithms for keyword extraction. Naïve Bayes [10, 26], decision trees [22] and SVM [30] are representative supervised methods, which "achieved satisfying accuracy and have excellent stability" [17], but require an annotated corpus. Moreover, it's difficult for supervised methods to extract unknown keywords [17]. Unsupervised algorithms include string frequency or feature weight calculation [8, 17, 23, 27] and semantic network structure analysis [15] methods. Previous keyword extraction algorithms are designed for single documents. Our system adopts and improves the unsupervised methods in the topic keyphrase extraction process.

3. Topic detection and tracking based on burstiness of terms and aging theory

The first stage of news topic keyphrase extraction is to organize news stories, published by various Websites, into topics online automatically. The topic generation algorithm used in this paper is the same as that in our previous work [25], so we just give a brief description here.

3.1. Story representation based on burstiness of terms

The burstiness of terms is calculated in a similar way to [20] and added in the story representation. The burstiness value of term w during time slot i is symbolized as $b_i(w)$.

Stories are represented using term vectors.

Incremental TF-IDF model is widely applied to term weight calculation in TDT [2, 5, 28, 29]. We choose this model as a base to weight terms. *DF* (document frequency) of term w in time slot i is calculated as:

$$df_i(w) = df_{i-1}(w) + df_{S_i}(w) \quad (1)$$

where S_i means a set of stories coming during time slot i, and $df_{Si}(w)$ means the number of stories that term w appears in. $df_{i-1}(w)$ represents the number of stories that term w appears in before time slot i-1 (included). A training corpus comprised of a sufficient amount of stories is used for the calculation of *DF* initially. As shown in formula (1), *DF* is updated dynamically in each time slot i.

Then each story d in time slot i is represented as an n-dimension vector, where n is the number of distinct terms in story d. Each dimension is weighted using a combination of incremental TF-IDF model and B-VSM model [14], which considers the burstiness of terms. And the vector is normalized so that it is of unit length:

$$weight(d,w) = \frac{tf(d,w)\log((N_i+1)/(df_i(w)+0.5))\cdot b_i(w)}{\sqrt{\sum_{w'\in d}(tf(d,w')\log((N_i+1)/(df_i(w')+0.5))\cdot b_i(w'))^2}} \quad (2)$$

where $tf(d, w)$ means how many times term w appears in story d and N_i represents the total number of stories before time slot i (included).

Cosine similarity is used to calculate the similarity between two stories.

3.2. Topic detection and tracking based on aging theory

Chen *et al* applied an aging theory to model a news topic's life span and considered a news topic as "a life form with stages of birth, growth, decay and death" [7]. They used the concept of energy function to track the life cycles of topics. The value of energy function indicates the liveliness of a news topic in its life span. The energy of a topic increases when it becomes popular and decreases as its popularity decays. Like in the nutriology, things that contribute to the energy of topics (e.g. new stories inserted into the topics) are called *nutrition*.

We combine our TDT algorithm proposed in [24] with the aging theory to perform online topic detection and tracking. A summary of the algorithm is given as follows:

New coming stories are clustered into new topic candidates, using the state-of-the-art clustering algorithms [31]. The similarity between topics t_1 and t_2 is calculated as the arithmetic average of pair-wise similarities between the stories in t_1 and the stories in t_2. A similarity threshold ($\text{threshold}_{\text{track}}$) is used to decide whether a new topic candidate is really new or not, after it has been compared with all previous "alive" topics. A new topic is generated if the candidate is not combined with any previous topic.

According to the aging theory, the energy of a topic increases when new stories are added to the topic, and decreases as time goes by. If the energy value is below a threshold, the topic is considered "dead" and removed in order to keep all topics in the system up to date. We make use of three functions from [7] to calculate and update the energy of topics in every time slot:

- *getNutrition()* calculates the nutrition that topic t receives from story d:
$$\alpha_m * \text{sim}(t, d) \quad (3)$$
where *sim()* represents the cosine similarity calculation function and α_m is a coefficient.
- *energyFunction()* converts a topic nutrition value into an energy value. A sigmoid function is adopted in this paper, analogous to that used in [7]:
$$energyFunction(x) = \begin{cases} \frac{x}{1+x} & x > 0 \\ 0 & otherwise \end{cases} \quad (4)$$
It is easy to see that $0 \leqslant energyFunction(x) < 1$.
- *energyDecay()* carries out the energy decrease in each time slot. The energy value of every topic is reduced with a decay factor β_m in every time slot. When no or few stories are added to a topic, its energy value will gradually decline. If the energy value is below β_m, the topic is considered "dead" and removed.

The parameters ($\text{threshold}_{\text{track}}$, α_m and β_m) are determined by experiments with training data.

4. News topic keyphrase extraction

News stories are mostly short. It was observed in [15] that semantic network structure analysis method performed relatively poor for short documents because the relation between nodes was sterile and "most nodes made no differences in a semantic network, while in TF-IDF might do". Our algorithm is a combination of feature weight and string frequency calculation. Unlike previous methods, our system firstly selects keyword candidates from each news story according to their weights, and then performs a phrase identification process to combine adjacent keywords into keyphrases, together with some surrounding words if necessary. In this way, we avoid comparing the weights between words and phrases of different lengths.

4.1. Generation of keyword candidates in a news story

Each term in story d is weighted using IDF, burstiness and position information. The position weight of term w is calculated as:

$$weight_p(w) = \sum_i score(p_i(w)) \quad (5)$$

where $p_i(w)$ is an occurrence position of term w, and $score(p_i(w))$ is defined as:

$$score(p(w)) = \begin{cases} 7 & p(w) = title \\ 3 & p(w) = FirstParagraph \\ 1 & otherwise \end{cases} \quad (6)$$

Formula (6) is revised from a formula in [23]. We remove the weighted score for occurrence positions in the last paragraph because we find them not so important from observation.

Finally, term w is weighted as:

$$weight(w) = weight_p(w) \cdot \log((N_i + 1)/(df_i(w) + 0.5)) \cdot b_i(w) \quad (7)$$

where N_i, $df_i(w)$ and $b_i(w)$ are defined in Section 3.

Ten terms with the highest weight are generated as keyword candidates of each news story. This empirical number of terms is large enough to keep useful information of the story, according to our observation and previous work [17, 23]. Keyword candidates of all stories in topic t form a keyword candidate set of t, symbolized as KCS_t.

4.2. Filter of keyword candidates using topic information

Every keyword candidate w of each story in topic t is further filtered by the number of times w is generated as a keyword candidate of single stories. The threshold is set as $T_{filter} = \alpha_k * N_t$, where N_t is the number of stories in t and $\alpha_k = 0.5$. So terms selected as keyword candidates by

fewer than half of all stories in t are abandoned. In this way, a large number of noisy candidates are successfully removed. For example, a term misspelt by a news editor might get high IDF and burstiness value and it's possible to select it as a keyword candidate. Terms of this kind can easily be filtered using topic information. The keyword candidate set after filtering is symbolized as KCS_2.

On the other hand, some good keyword candidates may have been deleted in the process above. We make use of co-occurrence information to "find them back". Keyword candidates, appearing in the same sentences with other candidates that are not filterd by T_{filter}, are recovered and combined with KCS_2. This new keyword candidate set is symbolized as KCS_3.

4.3. Phrase identification

Phrases are more complete than words in syntactic and semantic structures. In other words, keyphrases have clearer meanings than keywords and thus are better representations for topics. So a phrase identification process is performed to combine and transform keyword candidates into keyphrase candidates. Moreover, since errors exist in word segmentation when dealing with Chinese texts, a keyword may have been split into several successive smaller words or characters. A phrase identification process will merge these smaller words into the correct and complete word and further combine words into phrases.

All keyword candidates in KCS_2 are firstly put into the phrase candidate set *PCS*. Then keyword candidates in KCS_3 are combined into phrases and put into *PCS*, if their occurrence positions in the news stories are adjacent. Furthermore, the adjacency relation is loosened to allow one word out of KCS_3 exists and more phrases are put into *PCS*.

There are several rules for phrase identification. First, the phrase p must be a max-duplicated string whose frequency is larger than that of all strings containing p [27]. Second is a significance estimation function from [8]:

$$SE_p = MI_{ab} = \frac{f_p}{f_a + f_b - f_p} \quad (8)$$

where p is the phrase to be identified, $p=c_1c_2...c_n$ (c_i is a character), a and b are the two longest composed substrings of p with the length n-1, i.e., $a=c_1c_2...c_{n-1}$ and $b=c_2c_3...c_n$, f_p, f_a and f_b are the frequencies of p, a and b. We remove phrases with *SE* value smaller than the threshold T_{SE}, which is determined by experiments with training data. All frequencies mentioned above are calculated within a topic. These rules for phrase identification were used for keyword extraction from single documents previously. They are more effective when used for phrase identification within a topic, which is composed of many related stories.

4.4. Topic keyphrase selection

Remaining phrases in *PCS* after the phrase identification process are ranked by the following rules, sequentially: (1) phrases with more keyword candidates in KCS_2 are ranked higher; (2) the weight of phrase p is calculated as the largest weight of all keyword candidates in KCS_2 that p contains, and phrases with larger weights are ranked higher; (3) more frequent phrases are ranked higher; (4) shorter phrases are ranked higher.

Topic keyphrases are finally selected from the ranked phrase sequence in *PCS*. In order to avoid redundancy, if more than half of the keyword candidates contained in a phrase have been covered by selected keyphrases, the phrase will not be adopted. The total number of selected topic keyphrases is N_p, which can be adjusted according to practical need.

5. Experiments

Preliminary experiments are firstly performed on a training dataset to find proper values for parameters. Then the results of automatic online news topic keyphrase extraction are presented.

5.1. Dataset and experimental setup

Experiments are performed on datasets constructed from practical Web environment. Crawlers are gathering news pages from dozens of Chinese news Websites all the time. The collected news stories are filtered using a keyword list to get only stories about search engine related companies, such as Google, Yahoo! and etc. We focus on the search engine related domain because: (1) we're familiar with the domain; (2) the proposed system is initially designed for watching topics of the search engine related domain online. In fact, it is not necessary to consider datasets with multi-domain news because news stories of various categories are usually put in corresponding channels of news Websites.

News stories published from Jan 1 to Oct 31, 2007 are studied in this paper. There are 53,369 stories in total, divided into two parts:

- TrainingSet: contains 14,602 news stories, published from Jan 1 to Mar 31, 2007
- TestingSet: contains 38,767 news pages, published from Apr 1 to Oct 31, 2007

Topic keyphrase extraction results are evaluated in terms of precision (p), recall (r) and F-measure ($2*p*r/(p+r)$).

5.2. Parameter settings

Preliminary experiments are performed on TrainingSet to find proper values for parameters: $\text{threshold}_{\text{track}}$, α_m, β_m and T_{SE}. Topics are generated and updated using traditional TDT algorithms from [24].

50 topics with keyphrases are labeled by assessors. The TDT results on the labeled topics are best with $\text{threshold}_{\text{track}}$=0. 182. α_m and β_m are calculated as the method used in [7]: For each topic, a proportion r_1 of the total nutrition corresponds to a proportion s_1 of the total energy. By using two points (r_1, s_1) and (r_2, s_2), α and β can be solved. We use the averages of α_m and β_m of 50 topics as the final parameter values: α_m=0.251031, β_m= 0.001793. There are different demands of *SE* value on phrases of different lengths: for phrases length=2, the threshold T_{SE}= 0.42; for phrases length=3, T_{SE} =0.73; for phrases length=4, T_{SE} =0.78; for others, T_{SE}= 0.86.

5.3. Keyphrase extraction results

TestingSet is used to perform the automatic online topic keyphrase extraction experiment. The length of time slots is 15 minutes. The topic detection and tracking algorithm described in Section 3 is performed. Keyphrase extraction process is performed on newly created or updated topics. Table 1 below shows the keyphrases extracted by the proposed algorithm. The results of top 3 topics are listed here. Topics are ranked following the algorithm in [25].

Table 1. Keyphrases extracted from top 3 topics on search engine related companies, 8:00 a.m., Oct 26, 2007

Topic	Some of the extracted keyphrases
The upcoming Alibaba IPO in Hong Kong	阿里巴巴香港 IPO(Alibaba IPO in Hong Kong), 散户的公开认购(private investors' offer for subscription), 马云(Jack Ma), …
Baidu to enter C2C E-commerce Market	百度宣布进军 C2C 电子商务(Baidu announces to enter C2C E-commerce Market), 搜索引擎巨头(a titan of search engines), …
Microsoft invests $240 million in Facebook	微软入股 facebook(Microsoft invests in Facebook), 网络广告市场(Web advertisement market), 雅虎和 google(Yahoo! and Google), …

The results of 50 randomly selected topics are evaluated by assessors. Keywords in KCS_1 (see Section 4.1) are ranked by their weights and used as a baseline result, after redundancy removal. Figure 1 below shows the performance:

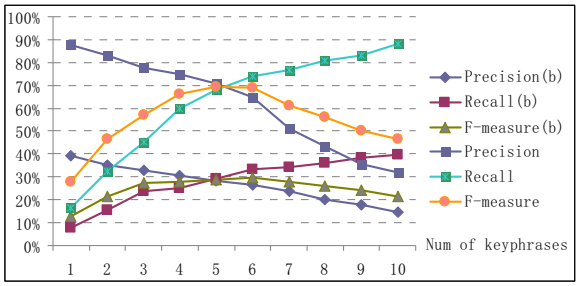

Figure 1. Performance of topic keyphrase extraction when various numbers of keyphrases are extracted ((b) represents the baseline result)

From figure 1, our method with keyword candidate filtering, phrase identification, keyphrase ranking and selection has greatly improved the performance from the baseline result.

6. Conclusions and future work

In this paper, we propose an automatic online news topic keyphrase extraction system. The proposed system extracts keyword candidates from single news stories, filters them with topic information and then combines them into phrase candidates using position information. Finally, the phrases are ranked and top ones are selected as topic keyphrases. Related news stories of topics are provided for users' quick access if they are interested after viewing the topic keyphrases. Empirical evaluation on experimental results indicates that the proposed system works effectively, with a performance of precision=70.61% and recall=67.94%.

In the future, we plan to analyze the changes of topic keyphrases when topics are updated. We are also interested in studying the effect of term relationship, which is represented as keyphrases of the same topics, on Web information retrieval results.

7. Acknowledgements

This work is supported by the Chinese National Key Foundation Research & Development Plan (2004CB318108), Natural Science Foundation (60621062, 60503064, 60736044) and National 863 High Technology Project (2006AA01Z141). The authors would like to thank Bin Liang and Zaihong Qu for their work on evaluation and demonstration. They also thank the anonymous reviewers for their useful comments.

8. References

[1] http://www.nist.gov/speech/tests/tdt/

[2] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In SIGIR98, 37-45.

[3] J. Bai, J.-Y. Nie, H. Bouchard and G. Cao. Using Query Contexts in Information Retrieval. In SIGIR2007:15-22.

[4] J. Bai, J.-Y. Nie and G. Cao. Context-Dependent Term Relations for Information Retrieval. In EMNLP, pages 551-559, 2006.

[5] T. Brants, F. Chen, and A. Farahat. A System for New Event Detection. In SIGIR2003, 330-337.

[6] G. Cao, J.-Y. Nie and J. Bai. Integrating Word Relationships into Language Models. In SIGIR2005:298-305.

[7] C.C. Chen, Y.T. Chen, Y. Sun and M.C. Chen. Life Cycle Modeling of News Events Using Aging Theory. In Proceedings of 14th European Conference of Machine Learning (ECML '03), pp. 47-59, 2003.

[8] L.-F. Chien. PAT-tree-based Keyword Extraction for Chinese Information Retrieval. In SIGIR97:50-58.

[9] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan. UMass at tdt 2004. In 2004 Topic Detection and Tracking Workshop (TDT'04), 2004.

[10] E. Frank, G. Paynter, I. Witten and et al. Domain-specific keyphrase extraction. In IJCAI, 1999:668-673.

[11] G.P.C. Fung, J.X. Yu, H. Liu and P.S. Yu. Time-Dependent Event Hierarchy Construction. In Proceedings of KDD2007, pages 300-309, California, USA, 2007.

[12] G.P.C. Fung, J.X. Yu, P.S. Yu and H. Liu. Parameter free bursty events detection in text streams. In Proceedings of the 31st VLDB Conference, pages 181-192, Trondheim, Norway, 2005.

[13] Q. He, K. Chang, and E. P. Lim. Analyzing Feature Trajectories for Event Detection. In SIGIR2007, 207-214.

[14] Q. He, K. Chang and E. P. Lim. Using Burstiness to Improve Clustering of Topics in News Streams. In Proceedings of the 7th IEEE International Conference on Data Mining, pp. 493-498, 2007.

[15] C. Huang, Y. Tian, Z. Zhou and et al. Keyphrase Extraction using Semantic Networks Structure Analysis. In ICDM 2006.

[16] G. Kumaran and J. Allan. Text Classification and Named Entities for New Event Detection. In SIGIR2004, 297-304.

[17] J. Li, Q. Fan and K. Zhang. Keyword Extraction Based on tf/idf for Chinese News Document. Wuhan University Journal of Natural Sciences, 2007, 12(5):917-921.

[18] M. Spitters and W. Kraaij. TNO at TDT2001: Language Moel-Based Topic Detection. Topic Detection and Tracking Workshop Report, 2001.

[19] N. Stokes and J. Carthy. Combining Semantic and Syntactic Document Classifiers to Improve First Story Detection. In SIGIR2001, 424-425.

[20] R. Swan and J. Allan. Automatic Generation of Overview Timelines. In SIGIR2000, 49-56.

[21] D. Trieschnigg and W. Kraaij. Hierarchical topic detection in large digital news archives. In Proceedings of the 5th Dutch Belgian Information Retrieval workshop, 2005.

[22] P. Turney. Learning algorithms for keyphrase extraction. Information Retrieval, 2000, 2(4):303-336.

[23] H. Wang, S. Li and S. Yu. Automatic Keyphrase Extraction from Chinese News Documents. In FSKD 2005, LNAI 3614: 648-657.

[24] C. Wang, M. Zhang, S. Ma and L. Ru. Automatic online news issue construction in Web environment. In proceedings of the 17th international conference on World Wide Web, 2008, 457-466.

[25] C. Wang, M. Zhang, L. Ru and S. Ma. Automatic Online News Topic Ranking Using Media Focus and User Attention Based on Aging Theory. *To appear* in Proceedings of the 17th Conference on Information and Knowledge Management, 2008.

[26] I. Witten, G. Paynter, E. Frank and et al. KEA: Practical Automatic Keyphrase Extraction. In Proceedings of the fourth ACM conference on Digital libraries, 1999:254-255.

[27] W. Yang and X. Li. Chinese Keyword Extraction Based on Max-duplicated Strings of the Documents. In SIGIR2002:439-440.

[28] Y. Yang, T. Pierce, and J. Carbonell. A Study of Retrospective and On-line Event Detection. In SIGIR98, 28-36.

[29] K. Zhang, J. Li, and G. Wu. New Event Detection Based on Indexing-tree and Named Entity. In SIGIR2007, 215-222.

[30] K. Zhang, H. Xu, J. Tang and et al. Keyword Extraction Using Support Vector Machine. In Proceedings of WAIM, 2006:85-96.

[31] Y. Zhao and G. Karypis. Criterion Functions for Document Clustering. Technical Report, 2005.