News Keyword Extraction for Topic Tracking

Sungjick Lee, Han-joon Kim†
Department of Electrical and Computer Engineering, University of Seoul, Korea
sjleekor@gmail.com, khj@uos.ac.kr

Abstract

This paper presents a keyword extraction technique that can be used for tracking topics over time. In our work, keywords are a set of significant words in an article that gives high-level description of its contents to readers. Identifying keywords from a large amount of on-line news data is very useful in that it can produce a short summary of news articles. As on-line text documents rapidly increase in size with the growth of WWW, keyword extraction has become a basis of several text mining applications such as search engine, text categorization, summarization, and topic detection. Manual keyword extraction is an extremely difficult and time consuming task; in fact, it is almost impossible to extract keywords manually in case of news articles published in a single day due to their volume. For a rapid use of keywords, we need to establish an automated process that extracts keywords from news articles. We propose an unsupervised keyword extraction technique that includes several variants of the conventional TF-IDF model with reasonable heuristics.

1. Introduction

Recently, a Korean research agency has reported that more than half of Internet users enjoy the news service provided by Internet portal sites [2]. One of Korean major portal sites, *Naver* (http://www.naver.com) provides more than 10,000 news articles per day. However, it is not easy for users to grasp all of news articles shortly. Normally, to help user to easily browse many articles, the hottest news articles tends to be presented in the top pages of news portal sites. In this regard, keywords extracted from a huge numbers of news can be very helpful for browsing them effectively.

Keyword extraction technique is used to extract main features in studies such as information retrieval, text categorization, topic detection, and document summarization. To extract keywords, *TF-IDF* (Term Frequency-Inverse Document Frequency) weighting model has been widely used. However, this weighting model is a statistical model that evaluates the degree of importance of a word in a single document. In contrast, for keyword extraction, we need to consider a weighting scheme in a whole document collection level. We propose a number of variants of the conventional *TF-IDF* model for keyword extraction. With these variants, we extract keywords for each news domain (e.g., politics, business, society, entertainment and so forth). Moreover, we propose cross-domain filtering for stop-word removal and a new measure for term frequency, called table term frequency (*TTF*). These extracted keywords through these variants of *TF-IDF* and cross-domain filtering are highly useful for summarizing a huge number of news articles in each news domain. Also, keywords can be utilized usefully for news article classification and news exploration.

2. Problem definition

The problem to be solved in this paper is to extract significant keywords for each news domain. Let $D = \{d_1, d_2, \ldots, d_m\}$ be a set of news documents that belong to each news domain. And let $T_j = \{t_{j1}, t_{j2}, \ldots, t_{jn}\}$ be a set of n terms extracted from a single news document d_j. Then T, a set of terms extracted from a document set D is a union of $T_1, T_2, \ldots, T_m$. To extract significant keywords from T, we need to weight terms of T. Also, we should sort the terms of T by this weight and extract top-N terms with high weight. With these terms, we make a word list named 'candidate keyword list'. However, this list may include some meaningless words that appear too frequently in news documents, such as 'editor' and 'news'; such a word can be regarded as a stop-word. In our work, by removing the stop-words through cross-domain filtering, we make a final keyword list, called 'keyword list'.

† Correspondence: Han-joon Kim, Dept. of Electrical and Computer Engineering, Univ. of Seoul, 90 Jeonnong-dong Dongdaemun-gu Seoul, Korea

978-0-7695-3322-3/08 $25.00 © 2008 IEEE
DOI 10.1109/NCM.2008.199

3. Keyword extraction

In this paper, we propose a two-stage keyword extraction system. In the first state, we extract candidate keywords from a given document set. Next, in the second stage, we remove meaningless words by comparing candidate keywords in terms of ranking results.

3.1. *TF-IDF* weight and variants

TF-IDF weighting model has been widely used in information retrieval and text mining. This weighting is a statistical measure used to evaluate how important a particular word is in a document. However, we want to know how important a word is within a whole document collection. Regarding *TF* as a degree of importance within a whole document collection, we make several *TF-IDF* weights variants. First, we explain conventional *TF-IDF* weight scheme.

Table 1. Conventional TF-IDF weight

TF	$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$ $n_{i,j}$: the number of occurrences of the considered term in document d_j $\sum_k n_{k,j}$: the number of occurrences of all term in document d_j
IDF	$idf_i = \log \frac{\lvert D \rvert}{\lvert \{d_j : t_j \in d_j\} \rvert}$ $\lvert D \rvert$: total number of documents in the corpus $\lvert \{d_j : t_j \in d_j\} \rvert$: number of documents where the term t_i appears
TF-IDF	$tfidf_{i,j} = tf_{i,j} \times idf_i$

3.1.1. TF-IDF weight. *TF-IDF* value is composed of two components: *TF* and *IDF* values. The rationale of *TF* value is that more frequent words in a document are more important than less frequent words. *TF* value of a particular word in a given document is the number of occurrences in the document. This count is usually normalized to prevent a bias towards longer documents to give a measure of the importance of the term t_i within the particular document d_j, like a TF equation given in table 1. The second component of TF-IDF value, IDF, represents rarity across the whole collection. This value is obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient, like a IDF equation given in table 1. For example, a word, 'today' appears in many documents and this word is weighted as low IDF value. Thus it is regarded as a meaningless word. [5] shows theoretical justification for IDF.

3.1.2. TF-IDF weight variants. In this paper, we need to evaluate the degree of importance of a given word in a whole document collection, not in a document.

a. TF variants. For keyword extracting, we consider three TF variants (BTF, NTF1, NTF2) presented in figure 1. The first TF variant, BTF, means basically a frequency that is how often a given term occurs in the whole document collection.

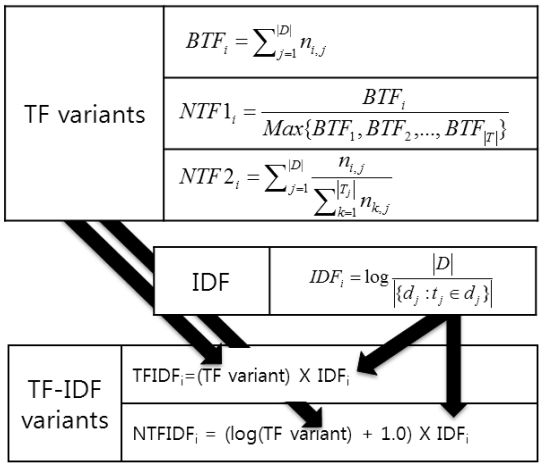

Figure 1. TF variants, IDF, TF-IDF variants

The BTF is obtained by summing up each count that a given term appears in each document. However, because this TF value is a simple count in whole document collection, this value can be some biases. Thus, this value needs to be normalized. With this reason, we propose other two normalized term frequency equation (NTF1, NTF2). The first bias of BTF that we need to be removed is a bias towards TF value is even larger than IDF value. To remove this bias, we propose the first variant that normalized by the maximum BTF in a given document collection. The second bias of BTF is that the words appeared in a long article may have larger frequency and may be regarded as more important words. Hence, we want to reduce such weight of document's length, which results in the second normalized TF, NTF2. The TF variant is to sum up the results of dividing 'the number of a given word appears in each document' by 'the number of all words appears in each document'. These three TF variants are used to obtain TF-IDF value.

b. TF-IDF variants. With three TF variants and an IDF equation used conventionally, we obtain TF-IDF values in two ways. The first one is to multiply a TF variant by IDF. But TF values obtained by BTF or NTF1 can be even larger than an IDF value because a IDF value is obtained by take logarithm of some value.

Thus, we propose the second way, NTFIDF in figure 1. NTFIDF is obtained taking logarithm of TF. But, if a TF value is lower than 1, a value taken logarithm is lower than 0. And this value is not proper for a weight to be assigned to a word. Thus, we add '1.0' to the value before multiplying it to an IDF value. After assigning two TF-IDF value to each words appeared in a document set, we take top N terms assigned as the highest TF-IDF values and make a 'Candidate keywords list' for each news domain.

① Calculating conventional TF-IDF for each document

date	document ID	term	TF-IDF
2008-06-18	15480	담화	15.787019
2008-06-18	15480	내일	10.756852
2008-06-18	15480	대통령	10.121913
2008-06-18	15480	사과 말씀	8.782413
2008-06-18	15481	담화	8.611553
2008-06-18	15481	청와대 비서진	8.581073
2008-06-18	15481	대통령	7.591434

② Calculating TTF (threshold for taking words : 0.6)

Total # of words in document 15480 : 4 ➡ 4 * 0.6 = 2.4
Candidate keywords : '담화', '내일'

Total # of words in document 15481 : 3 ➡ 3 * 0.6 = 1.8
Candidate keywords : '담화'

date	term	TTF
2008-06-18	담화	2
2008-06-18	내일	1

Figure 2. Calculating TTF

3.2. Table Term Frequency: TTF

In previous section, we introduce TF-IDF variants used for weighting the occurring terms. And, now, we propose a more precise but simple model for extracting keywords. This model is composed of two steps. The first step is weighting terms appeared in each document using the conventional TF-IDF of Table 1. In this step, we can know which term is more important in a given document. And the second step is collecting the most important n% terms from each document according to TF-IDF value calculated in the first step. Then we count term frequencies from the terms collected. We named this term frequency as 'Table Term Frequency' because the terms collected are stored in a temporary table. We present an example in Figure 2. There are two documents, 15480 and 15481. The terms, '담화', '내일', '대통령' and '사과 말씀', occurs in document 15480. And the terms, '담화', '청와대 비서진' and '대통령' in document 15481. And we use a threshold for collecting terms as 0.6 that means choosing 60% terms as most. By this threshold, '담화' and '내일' are collected from the document 15480, and '담화' is collected from the document 15481. These collected terms are inserted to a temporary table and term frequencies for each term is counted. Finally, '담화' is the most important term than other terms that occurs in document set.

3.3. Cross-domain comparison filtering

A set of news documents for each domain is subset of a set of general news documents. Because this reason, although we assign weights to words of a document set for each news domain, the words that are important to 'general' news documents. These words are not proper as keywords of each news domain. And, if the keywords extracted from each news domain is tightly associated with each news domain and keyword sets for each news domain can be distinguished clearly, common keywords should not appear in news domains. Thus, in this paper, we improve the accuracy of extracted keywords by controlling appearance of same words as keywords. We propose a cross-domains comparison technique in following. We calculate a standard deviation with given word's ranks of news domains. If calculated standard deviation is less than some threshold, the given word is regarded as one to be removed. The reason of limiting with a standard deviation is that if a word ranked very high in one news domain and low in the other news domain, the word should be regarded as a important word in the news domain that it ranked high. In following figure 2, we show an example for calculating a standard deviation.

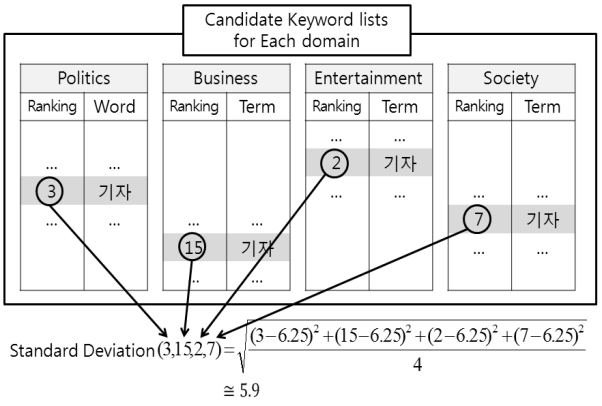

Figure 3. Calculating a standard deviation

To choose a word to be removed from 'Candidate keyword list' of each news domain, at first, we pick words that appeared in 'Candidate keyword lists' of all news domains. In figure 2, the Korean word '기자', a

journalist in English, ranked as third, fifth, second and seventh in each news domain. And the standard deviation for these rankings is about '5.9'. If the threshold for choosing meaningless words is determined as '10', the word '기자' is regarded as one to be removed. We remove these words and finally make 'Keyword lists' for each news domain.

4. Experiments

In this paper, we propose keyword extracting system for news documents with TF-IDF variants and cross-domain comparison. And we found a meaningful keyword list for each news domain.

4.1. Experiment environment

The first step to extract keywords is downloading news documents from Internet portal site, Naver. We use HTTP protocol implementation with JAVA language to download HTML files on web server. And we parse those HTML files to pure texts. The parsed news documents are processed with natural language processing module. And the results are stored in relational database.

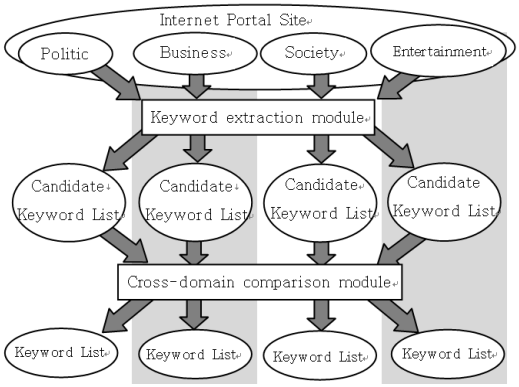

Figure 4. The architecture of keyword extraction system

4.2. Keyword Extraction system

The architecture of keyword extraction system is presented in figure 3. HTML news pages are gathered from a Internet portal site. And candidate keywords are extracted throw keyword extraction module. And finally keywords are extracted by cross-domain comparison module.

Keyword extraction module is described in detail. We make tables for 'document', 'dictionary', 'term occur fact' and 'TF-IDF weight' in relational database. At first the downloaded news documents are stored in 'Document' table and nouns are extracted from the documents in 'Document' table.

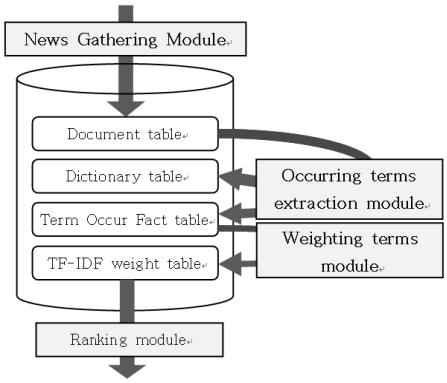

Figure 5. Keyword extraction module

Then the facts which words are appeared in documents are updated to 'Term occur fact' table. Next, TF-IDF weights for each word are calculated using 'Term occur fact' table and the result are updated to 'TF-IDF weight' table. Finally, using 'TF-IDF weight' table, we make 'Candidate keyword list' for each news domain with words ranked high.

4.2.1. Extracting news documents from Web Pages. Internet portal site, *Naver*, provides news pages by domain, such as flash, politics, business, society, life/culture, world, IT/science, entertainment, column, English, magazine and special. We choose four domains (politics, business, society and entertainment) to experiment. About 2000 news documents for politics, 6000 documents for business, 4000 documents for society and 1200 documents for entertainment are appeared every day. The news pages of HTML is written in a fixed structure. Using this structure of HTML page, we extract pure news article and metadata about the news (such as reported time, title and publisher). And extracted information is stored in database.

4.2.2. Data model and Database We make tables in relational database with following data model

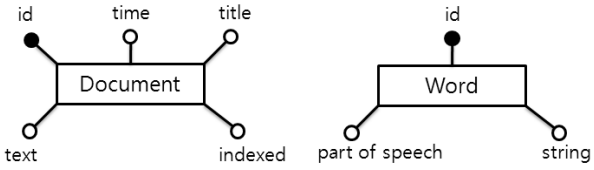

Figure 6. Data schema for document and word

a. Data model in relational database. In figure 5, models for news documents and extracted words are described. News document entity has properties such as 'id', 'time', 'title', 'text' and 'indexed'. The 'id' property means an identification number for a given document. And the 'time' property is the date that the

document appeared in the web page. Also, the 'title' and 'text' property is the title and body of a given news document. Specially, the 'indexed' property is added to store Boolean value indicating that nouns are extracted from the document. And word entity has three properties such as 'id', 'string', 'part of speech'. The first property 'id' is an identification number for a given word. And the 'string' property is used to store the word. The last property 'part of speech' is the part of speech of the word.

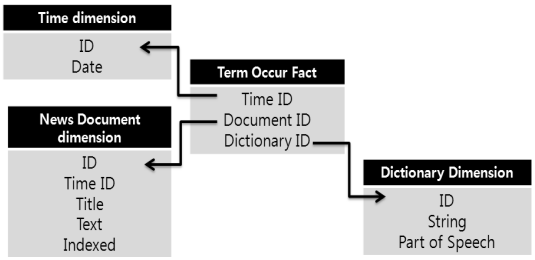

Figure 7. Star schema for term occurrence fact

b. A model for term occurrence fact table. To calculate TF-IDF weights for each extracted word, the facts which a given word is occurring in documents are stored in 'term occur fact' table. This table is made as same as Star schema used in data warehouse. Each row of 'Term occur fact' table references the primary keys of time dimension, news document dimension and dictionary dimension as seen in figure 6. Time, news document, and dictionary dimensions are implemented to time, document and dictionary tables in the proposed system. Fact table is used to record individual events over time. Thus, in the keyword extraction system, once a word is appeared in a document, IDs of time, the word and the document are updated to 'Term occur fact' table as one record. For example, if a word 'politician' appears 3 times in a given document and a word 'congress' appears 2 times in the document, total 5 records are updated to 'Term occur fact' table.

5. Experiment results

TFIDF			
Rank	**BTF**	**NTF1**	**NTF2**
1	이명박 후보	이명박 후보	서울
2	후보	후보	**무단전재**
3	**기자**	**기자**	한국언론뉴스허브
4	에리카 김	에리카 김	뉴시스통신사
5	**무단전재**	**무단 전재**	**재배포 금지**
6	**광고**	**광고**	모바일 연합뉴스
7	서울	서울	**재배포금지**
8	**재배포 금지**	**재배포 금지**	**저작권자연합뉴스**
9	검찰	검찰	오전
10	이명박	이명박	오후

NTFIDF (Normalized TFIDF with logarithm)			
1	이용득 위원장	이명박 후보	백승렬
2	이석행 위원장	에리카 김	생각하십
3	이회창 씨	검찰	권주훈 기자
4	금민 후보	BBK	패널 질문
5	고 팀장	김경준	대통합민주신당 정동영 대선후보
6	망 설치 창원	이명박	이광호기자
7	준 뉴스앤조이	이전	창당 10 주년 행사
8	금액 조성	기자회견	남강호기자
9	공기 음이온 보충	**재배포 금지**	박주성기자
10	초판 발행	서울	박지호

Table 2 Candidate keyword list for political news (Meaningless words are denoted as bold)

5.1. Candidate keyword lists for political domain

In this paper, we utilize six TF-IDF variants to find meaningful keywords for four news domain (politics, business, society and entertainment). To evaluate each TF-IDF variant, we experiment with political news appeared at November 11th, 2007. Table 3 shows the top 10 most heavily weighted terms by TF-IDF variants. The result derived by BTF and TF-IDF is better than others. Unfortunately, all results contain meaningless words.

5.2. Keyword List for each domain

Table 4 shows the top 10 terms of each domain with highest TF-IDF weight. Table 5 shows the top 10 'filtered' terms of each domain. The results before filtering contains many meaningless terms, but filtering with cross-domain comparison could remove many meaningless terms. Thus, keywords list in table 5 shows good results.

Table 3. Candidate keyword list for each domain (Words to be removed are underlined)

Rank	News Domain			
	Politics	**Business**	**Entertainment**	**Society**
1	이명박 후보	저작권자	제 28 회 청룡영화상 시상식	무단 전재
2	김경준	무단 전재	영화	뉴시스통신사
3	검찰	서울	레드카펫	재배포금지
4	한국언론뉴스허브	재배포 금지	해오름극장	한국언론뉴스허브
5	재배포금지	예정	재배포 금지	기자
6	뉴시스통신사	현재	청룡영화상	서울
7	오후	무단전재	보도자료	오후
8	저작권자연합뉴스	기자	시상식	저작권자연합뉴스
9	오전	내년	저작권자	재배포금지
10	재배포 금지	리얼타임뉴스	중구	모바일 연합뉴스

Table 4. Keyword list filtered with cross-domain comparison (Well-chosen words are denoted as italic.)

Rank	News Domain			
	Politics	**Business**	**Entertainment**	**Society**
1	*이명박 후보*	리얼타임뉴스	*제28회 청룡영화상 시상식*	*검찰*
2	*김경준*	아시아 경제	*영화*	방침
3	*검찰*	석간	*레드카펫*	지역
4	*이명박*	배포금지	*해오름극장*	혐의
5	*이면계약서*	멀티미디어	*청룡영화상*	지역 빛
6	*에리카 김*	경제뉴스	보도자료	독자희망
7	*계약서*	*기업*	*시상식*	부산일보사
8	*신당*	*외국인*	*장충동*	*이명박 후보*
9	*민주당*	*주가*	*국립극장*	*한나라당*
10	*김경준 씨*	*증시*	포즈	*김경준*

Table 5. Results of filtering on several thresholds (Bold words are removed when the threshold increases.)

Rank	Threshold for standard deviation			
	10	**100**	**1000**	**10000**
1	노 대통령	노 대통령	노 대통령	노 대통령
2	당선자	**당선자**	이명박 당선자	이명박 당선자
3	**광고**	이명박 당선자	인수위	인수위
4	이명박 당선자	**대통령**	**이명박 대통령 당선자**	노무현 대통령
5	대통령	**정부**	노무현 대통령	국회
6	**연합뉴스**	인수위	국회	총선
7	**저작권자 연합뉴스**	**이명박**	총선	인수위원회
8	정부	**대표**	인수위원회	대통령직
9	**오전**	이명박 대통령 당선자	대통령직	이명박 정부
10	인수위	**국민**	이명박 정부	정책

Table 6 shows the results with various thresholds for filtering. We have conducted the proposed filtering technique with 10, 100, 1000 and 10000 of standard deviation threshold. The threshold 10 yields a few meaningless words and the threshold 10000 results in removing too many terms. Empirically, filtering with 100 or 1000 of threshold produces optimal number of keywords.

Table 6. Keyword list with TTF

Rank	News Domain			
	Politics	**Business**	**Entertainment**	**Society**
1	대통령	정부	영화	화물연대
2	의원	협상	드라마	정부
3	정부	화물연대	시청자	혐의
4	국민	쇠고기	식객	경찰
5	청와대	가격	MBC	검찰
6	쇠고기	기업	이산	파업
7	대표	시장	김선아	쇠고기
8	한나라당	가능성	최강칠우	협상
9	이명박 대통령	파업	작품	대통령
10	국회	대통령	방송	촛불집회

5.3. Keyword List with TTF

Table 7 shows the results derived with TTF model from news documents appeared from June 17, 2008 to June 23, 2008. From the terms, '쇠고기', '협상', '촛불집회', '화물연대' and '파업', we can derive the issues of 'Deal beef imports between Korea and U.S.' and 'Transportation strike in Korea'.

6. Conclusions and Future Direction

In this paper, we propose the keyword extracting technique that effectively summarize and present topics from enormous set of news documents appeared in Internet portal site. To extract keywords, we have introduced 6 *TF-IDF* variants and filtering keywords with cross-domain comparison. We can remove meaningless words successfully with cross-domain comparison. Furthermore, we propose more precise more for extracting keywords, *TTF*. These keyword extracting techniques can be used to extracting main features from specified document set and applied to topic detection or opinion mining. In the future, we will study feature extraction from subjective documents for opinion mining, which identifies user sentiments from user review documents.

7. Acknowledgements

This research was supported by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Advancement. (Grant number IITA-2008-C1090-0801-0031).

References

[1] National Internet Development Agency of Korea, "Survey on the Computer and Internet Usage, Executive Summary, 2007.

[2] Robertson, S. E., "Term specificity [letter to the editor]", *Journal of Documentation*, Vol. 28, 1972, pp. 164-165.

[3] Robertson, S. E., "Specificity and weighted retrieval [documentation note]", Journal of Documentation, Vol.30, No.1, 1974, pp. 41-46,

[4] Robertson, S. E., "The probability ranking principle in information retrieval", Journal of Documentation, Vol. 33, 1977, pp.294-304.

[5] Stephen Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF", Journal of Documentation, Vol. 60, No.5, 2004, pp 503-520.