News-Oriented Automatic Chinese Keyword Indexing

Li Sujian[1] lisujian@pku.edu.cn
Wang Houfeng[1] wanghf@pku.edu.cn
Yu Shiwen[1] Yusw@pku.edu.cn
Xin Chengsheng[2] csxin@peoplemail.com.cn

[1]**Institute of Computational Linguistics, Peking University, 100871**
[2]**The Information Center of PEOPLE'S DAILY, 100733**

Abstract

In our information era, keywords are very useful to information retrieval, text clustering and so on. News is always a domain attracting a large amount of attention. However, the majority of news articles come without keywords, and indexing them manually costs highly. Aiming at news articles' characteristics and the resources available, this paper introduces a simple procedure to index keywords based on the scoring system. In the process of indexing, we make use of some relatively mature linguistic techniques and tools to filter those meaningless candidate items. Furthermore, according to the hierarchical relations of content words, keywords are not restricted to extracting from text. These methods have improved our system a lot. At last experimental results are given and analyzed, showing that the quality of extracted keywords are satisfying.

1 Introduction

With more and more information flowing into our life, it is very important to lead people to gain more important information in time as short as possible. Keywords are a good solution, which give a brief summary of a document's content. With keywords, people can quickly find what they are most interested in and read them carefully. That will save us a lot of time. In addition, keywords are also useful to the research of information retrieval, text clustering, and topic search [Frank 1999]. Manually indexing keywords will cost highly. Thus, automatically indexing keywords from text is of great interests.

News is always the main domain that people pay a large amount of attention to. Unfortunately, only a small fraction of documents in this field have keywords. However, compared to unrestricted text, news articles are relatively easy to extract keywords from, because they have the following characteristics. Firstly, a news document is always short in length, and usually, only important words or phrases repeat. Secondly, as a rule, the purpose of news articles is to illustrate an event or a thing for readers. Then this kind of articles usually place more emphasis on some name entities such as persons, places, organizations and so on. Lastly, important content often occurs the first time in the title, or in the anterior part of the whole text, especially the first paragraph or the first sentence in every paragraph. These characteristics will help us in keywords indexing.

Several methods have been proposed for extracting English keywords from text. For example, Witten[1999] adopted Naïve Bayes techniques, and Turney[1999] combined decision trees and genetic algorithm in his system. These systems achieved satisfying results. However, they need a large amount of training documents with keywords, which are just what we are in need of now. For the Chinese language, some researchers adopt the structure of PAT tree and make use of mutual information to obtain keywords [Chien 1997, Yang 2002]. Unfortunately, the construction of PAT tree will cost a lot of space and time. In this paper, aiming at the characteristics of news-oriented arti-

cles, resources and techniques of current situation, we will introduce a simple procedure to index keywords from text. Section 2 will describe the architecture of the whole system. In section 3, we will introduce every module in detail, including how to obtain candidate keywords, how to filter out the meaningless items, and how to score possible keyword candidates according to their feature values. In section 4, experimental results will be given and analyzed. At last, we will end with the conclusion.

2 System Overview

Keyword indexing can also be called keyword extraction. The definition of a keyword is not restricted to one word in our conception. Here, a keyword can be seen as a Chinese character string, which might consist of more than one Chinese word. These character strings can summarize the content of the document they are in.

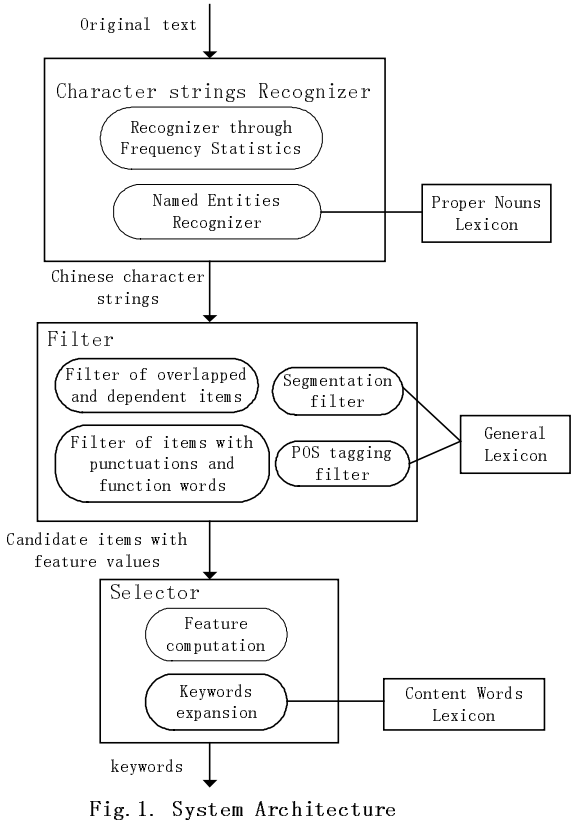

Fig. 1. System Architecture

Aiming at the task of keywords indexing, our system is designed and composed of three modules. As in figure 1, the first module is to recognize some Chinese character strings according to their frequency, and pick out those named entities in the text as the candidate keywords. The second module is a filter to remove all the meaningless character strings from the set of candidates. And the third module is a selector, which evaluates every candidate according to its feature values and choose from the candidate set those keywords with higher score. The higher score a character string has, the more content it will cover of the article it is in.

In our system, there are three kinds of lexicons. The lexicon of proper nouns is used to recognize named entities. The general lexicon includes Chinese words in common use, which is adopted for the segmentation and POS tagging of the text. And the lexicon of content words is used to expand the set of keywords. They will be introduced in detail in the following section.

3 System Design

3.1 Recognizer Module

It can be seen that one document is composed of a set of character strings. Every character string has its frequency in the document. In general, those character strings that occur several times can reflect the topic of the document. So, we take them out as keyword candidates. In addition, named entities, such as person names, place names, organizations, translation terms, titles of person and so on, are usually very important for the document without reference to their frequency. They will also be picked out from the text by named entities recognizer and input into the filter module with other character strings.

Unlike English, there are no explicit word boundaries in Chinese sentences, which makes it especially difficult to tell whether a character string is composed of one word or more than one word. Due to this characteristic, we don't use a dictionary, but get those character strings only according to their frequency statistics. We set a threshold value as 2 for the Chinese character strings considering the length of news documents. Suppose that a character string is $c_1c_2\ldots c_n$, and $f(c_1c_2\ldots c_n)$ represents its frequency, then we extract $c_1c_2\ldots c_n$ from text only if $f(c_1c_2\ldots c_n)$ equals to or is more than 2. That is, only a character string occurs two or more than two times, it can be selected as a candidate keyword.

There are two kinds of named entities. The first are those which have rules of composition, mainly Chinese names and foreign terms. They can be recognized with statistical and rule-based methods combined. Chinese names are composed of family names and first names, whose lengths are respectively 1 or 2 Chinese characters. Furthermore, there is a relatively stable set of family names, which often provide the anchor to search a name. For foreign terms, there are a relatively set of Chinese characters which are generally used as translation characters. Due to the limitation of the paper's length, we don't introduce the process of recognition in detail here. The other kind of named entities is mainly composed of proper nouns which represent names of places, organizations, person titles, etc. They often occur in news documents, but don't have rules of composition. Thus, we collect such words into our proper nouns lexicon. Then the module can find these named entities through looking up in this lexicon.

3.2 Filter Module

So far, Chinese character strings are generated only through frequency statistics. Thus, some of them stand out just because of simple repetition and are probably not meaningful units of language. We need to filter out those meaningless items. As in figure 1, we adopt four kinds of filters in filter module. They work as follows.

(1) Filter of Overlapped and Dependent Items

For two character strings S1 and S2, with S1 as a substring of S2, and the frequency of S1 is equal to that of S2, then S1 is overlapped by S2. In fact, we can set a threshold t_d for f(S1)-f(S2), where the function f(.) represents the frequency of some character string. If the value of f(S1)-f(S2) is less than t_d, then the string S1 is dependent on S2. Here, the overlapped and dependent substring will be removed from the candidate set.

(2) Filter of Items with Punctuations and Function words

The recognizer module treats equally all symbols in the text, such as Chinese characters and punctuations, etc. Thus when conducting the process of frequency statistics, for a character string, there might exist some punctuations and function words such as '。', '、', '了', '着', etc. These punctuations and function words usually occur in the head or tail of a character string. It's evident that such character strings can't serve as keywords of an article, and they should be deleted from the candidate set.

(3) Segmentation Filter

We find the first occurrence position of every candidate keyword and get the sentence at the position. Then the sentence is segmented. According to the segmented result, we can verify whether the character string is meaningful. First of all, we get the segmentation result of the character string in the segmented sentence. Suppose the character string $c_i \ldots c_j$ in the original text with the sentence $c_1c_2 \ldots c_{i-1}c_i \ldots c_jc_{j+1} \ldots c_n$ as its context, if the segmentation tool segments $c_{i-1}c_i$ or c_jc_{j+1} into one word, then $c_i \ldots c_j$ will not be regarded as an integrated unit. That is, this item will be seen as meaningless and filtered out from the set of candidate keywords. Here we don't adopt the method of conducting frequency statistics of words after segmentation, but use segmentation tool after frequency statistics of character strings. There are some reasons. Above all, although the segmentation technique is relatively mature, its precision is still not high enough. Then, for the same character string, its segmentation results often differ in different sentences. Thus, it's difficult to compute the frequency of a character string precisely. Furthermore, now we only need to segment one sentence for a candidate keyword. That will save us a great deal of time.

(4) POS Filter

Because keywords provide a brief summary for one document, they should be words or phrases that represent some meaning units such as nouns and noun phrases. Therefore, a single word whose part of speech is preposition, adverb, adjective, or conjunctive is filtered out. At the same time, verb phrases, adjective phrases, preposition phrases are also excluded from the candidate set. The same as segmentation filter, we only do the POS tagging for the sentence where every candidate keyword occurs the first time. If a candidate item is made of more than one word, it will have a sequence of POS tags according to which we can assign a phrase category. The POS tags or phrase categories are the basis for POS filtering.

Only conducting frequency statistics of character strings can't refine the candidate set well, and we utilize the relatively mature linguistic segmentation and POS tagging techniques so that we can further improve the quality of the candidate key-

words. Here, the general lexicon with about 60,000 Chinese words is applied to the processes of segmentation and POS tagging.

3.3 Selector Module

After several filtering, now we can get a reduced set of candidate keywords. Most character strings in the set are meaningful and reflect the content of the document to some extent. For every candidate now, we adopt several features to describe it. The features include frequency, length, position of the first occurrence, part of speech and whether it is a proper noun or in a pair of specific punctuations, as in table 1. At the same time, through the processing of several linguistic tools in filter module, we can assign a value to every feature in every candidate item.

feature	meaning of feature
freq	Frequency of an item
len	Length of an item
is_noun	Whether an item is a noun phrase
in_title	Whether the first occurrence of an item is in the title of one document
in_seg1	Whether the first occurrence of an item is in the first paragraph of one document
is_proper	Whether an item is a proper noun, for example: person name, organization, translation term, place name, title of a person etc.
in_sign	Whether an item is bracketed by a pair of specific punctuations such as ‘《》’ and ‘“”’.

Table 1. Features of candidate keywords

We can find that the candidate set is still too large to select from it the keywords. Then we will conduct feature calculation to refine the candidate set. We have known that every candidate item has a feature-value set. These feature values are our basis to evaluate every candidate item. We compute a score for every candidate keyword through the module of feature computation. The higher the score, the more relevant the candidate is to the document.

We compute the percentage how much manually indexed keywords of different lengths cover in the set of automatically generated candidates. As in figure 2, Length represents the length of keywords and percentage denotes the corresponding percentage that keywords of this length are in the set. The higher the percentage, the more likely the keywords of this length are to be selected. Therefore, we can make a conclusion that the score of a candidate is directly proportional to the percentage of its length. Then we can acquire the relation between score and length of a candidate. At the same time, we can also see that the score is directly proportional to a candidate's frequency. In addition, score is relevant to other features in table 1. Thus, we get formula 1, as following.

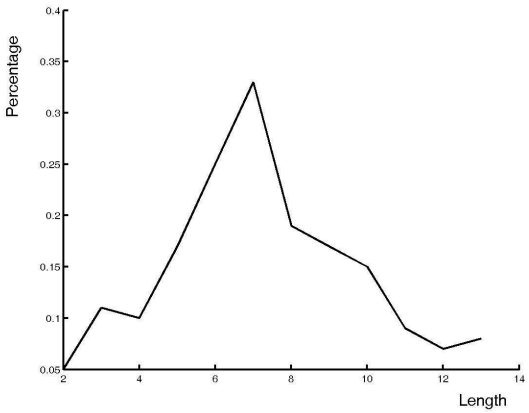

Fig. 2. Relations between Percentage Selected and Length of Keywords

$$score(ck) = Freq(ck) * \ln \frac{100}{(len(ck) - 7.1)^2} * \prod_{\text{fi}\in\text{F}} w_i^{\,f_i(ck)}$$

$$f_i(ck) = \begin{cases} 1 & \text{if ck satisfies the i}^{\text{th}}\text{ feature} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where ck represents a candidate keyword, the function freq(ck) gets the frequency of ck, len(ck) represents its length, that is, the number of Chinese characters every item includes. F represents all the binary features of a candidate keyword as in table 1. Every feature except the features of freq and len are denoted by f_i. $f_i(ck)$ is a binary function and its value is 0 or 1. If a candidate item ck satisfies the i^{th} feature, then the value is set to 1, otherwise, it's set to 0. w_i is the corresponding weight of feature f_i. For features is_noun, in_title, in_seg1, is_proper and in_sign, we set their weights to 7, 13, 5, 11 and 3 respectively by experience. After each candidate keyword gets a score, we choose those whose scores rank higher as keywords.

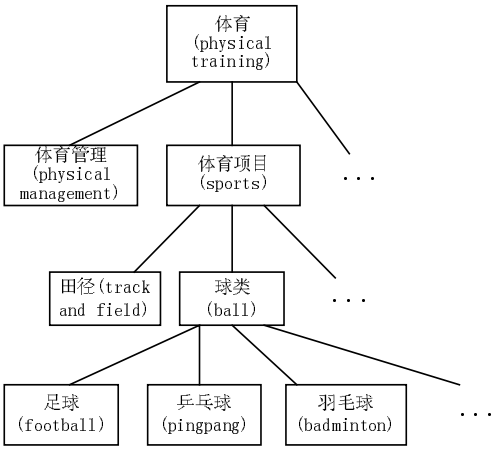

Fig. 3. A Sample Tree Structure of Content Words

Now the keywords we get are all selected from the original text. However, some keywords may express the content of the document, but they don't occur in the text. Therefore, we have constructed one list of content words with hierarchical relations as in figure 3. That is content words lexicon. The lexicon contains about 1,200 words which are often used as keywords. As the content words lexicon available now, we can look up in it and expand obtained keywords to a higher level, i.e., if a selected keyword has a parent in the lexicon, the parent word will be expanded as a keyword.

4 Experimental Results and Analysis

We select 37 news articles from China Daily as our testing material from which experts have manually extracted keywords. There are 23 articles about national politics, 10 articles of international politics, and 4 sports news articles. Here, we automatically extracted keywords from them and evaluated the results with the standard measures of precision and recall, which are defined as follows:

$$P = \frac{\text{numberof genuinekeywords recognized}}{\text{numberof keywordsindexingautomatically}}$$

$$R = \frac{\text{numberof genuinekeywords recognized}}{\text{numberof keywordsindexingmanually}}$$

Where P represents precision, and R represents recall. In general, these two measures in one system are opposite to each other. When precision is higher, recall will be lower. Otherwise, when precision is improved, recall will decrease. In table 2, we illustrate our experimental results. The first three rows give measures for articles about different styles and the figures in parentheses represent the number of articles. The fourth row gives the average measure of our system. For comparison, we also illustrate the results of Chien's [1997] PAT-tree-based method from his experiments in the last row. From this table, we can see that more emphasis is placed on precision in Chien's system. However, we incline to enhancing recall when precision and recall are assured relatively balanced. When precision is lower, perhaps more noise is introduced into the set of candidate keywords. Because we have adopted segmentation and POS tagging tools which can verify whether a candidate character string is a meaningful unit and found that the noise introduced now is more or less relevant to the content of the article, we don't have to worry more about precision. Therefore, we hope to generate more keywords automatically under the condition that the number of noise words is accepted.

	Recall	Precision
National politics (23)	0.452	0.401
International Politics (10)	0.644	0.594
Sports news (4)	0.629	0.482
Average	0.523	0.462
Chien's (exact match)	0.30	0.43

Table 2. Experimental Results

It has to be pointed out that there are no satisfactory results in extracting keywords from texts [Chien, 1997]. Although some keywords extracted are the same as manually extracted ones in meaning, they are often different due to one or two characters mismatched. According to our analysis of experimental results, though only 46% of extracted keywords appear in the set of manual keywords, the rest are also relevant to the text and adapt to the need of information retrieval. At the same time, about 52% of the manual keywords are generated by the automatically indexing method, however, we can often find a substitute for most of the rest in the set of automatically generated keywords.

Most of the keywords missed occur only once in the text, but they are mostly proper nouns of places, organizations or titles of person. And this

reveals that we need to further improve the techniques to recognize proper nouns.

5 Conclusion and Future Work

We have described a system for automatically indexing keywords from texts. One document is inputted into the recognizer module, the filter module and the selector module consecutively, with keywords output. Here we utilize the mature techniques available now such as string frequency statistics, segmentation and POS tagging tools. Then, according to features, we propose our method to evaluate directly every candidate keyword and select those with higher scores as keywords. At the same time, we break through the tradition of generating keywords only from the original text and acquire some keywords through looking up in the lexicon of content words with hierarchical relations. The experimental results show that our system can perform comparably to the state of the art.

Owing to the limit of the training corpus, the parameters in scoring formula are set by experience values. With our method, we can cumulate more and more documents with keywords. Then we can adopt machine-learning methods to conduct keyword indexing, which can make parameters more objective. That will be our further work.

References

[Chien 1997] Chien, L. F., PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval, Proceedings of the ACM SIGIR International Conference on Information Retrieval, 1997, pp. 50--59.

[Frank 1999] Frank E., Paynter G.W., Witten I.H., Gutwin C., and Nevill-Manning C.G., Domain-specific keyphrase extraction, Proc. Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, 1999, pp. 668-673.

[Lai 2002] Yu-Sheng Lai, Chung-Hsien Wu, Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology, ACM Transactions on Asian Language Information Processing (TALIP), Vol.1, No.1, March 2002, pp. 34-64.

[Liu 1998] Liu Ting, Wu Yan, Wang Kaizhu, An Chinese Word Automatic Segmentation System Based on String Frequency Statistics Combined with Word Matching, Journal of Chinese Information Processing, Vol.12, No.1, 1998, pp. 17-25.

[Ong 1999] T. Ong and H. Chen, Updateable PAT-Tree Approach to Chinese Key Phrase Extraction Using Mutual Information: A Linguistic Foundation for Knowledge Management, Proceedings of the Second Asian Digital Libaray Conference, Taipei, Taiwan, Novemeber 8-9, 1999.

[Turney 1999] Turney, P.D., Learning to Extract Keyphrases from Text, NRC Technical Report ERB-1057, National Research Council, Canada, 1999.

[Witten 1999] Witten I.H., Paynter G.W., Frank E., Gutwin C., and Nevill-Manning C.G., KEA: Practical automatic keyphrase extraction, Proc. DL '99, 1999, pp. 254-256.

[Yang 2002] Wenfeng Yang, Chinese keyword extraction based on max-duplicated strings of the documents, Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002, pp. 439-440.