News-Oriented Keyword Indexing with Maximum Entropy Principle*

Li Sujian[1] Wang Houfeng[1] Yu Shiwen[1] Xin Chengsheng[2]
[1]Institute of Computational Linguistics,
Peking University, 100871, Beijing, China
{lisujian, wanghf, yusw}@pku.edu.cn
[2] The Information Center of PEOPLE'S DAILY,
100733, Beijing, China
csxin@peoplemail.com.cn

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. Aiming at news documents' characteristics and the resources available, this paper proposes to use Maximum Entropy (ME) model to conduct automatic keyword indexing. The focus of ME-based keyword indexing is how to obtain all the candidate items and select useful features for ME model. First, we make use of some relatively mature linguistic techniques and tools to obtain all the possible candidate items. Then, a feature set of ME model will be introduced. At last we test the model, and experimental results are given.

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.

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. Due to the characteristics of 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 articles, resources and techniques available, we will introduce ME model to index keywords from text.

Section 2 will describe the architecture of the whole system and review the ME model. In section 3, we will introduce how to obtain candidate keywords as input of the ME model. In section 4, we will illustrate the process of how to construct a feature set for the ME model. In section 5, experimental results will be given and analyzed. At last, we will end with the conclusion.

2 System Architecture

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 might consist of more than one Chinese word, i.e., a term reflecting the main content of a document. In fact one document is composed of a set of terms every of which can be described by many features and must belong to a keyword or not. Thus, the

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1998030507-4

probability of a candidate being a keyword or not can be calculated through the maximum entropy model. According to the probabilities, we score for every candidate and select those with higher scores. The system is designed as in figure 1.

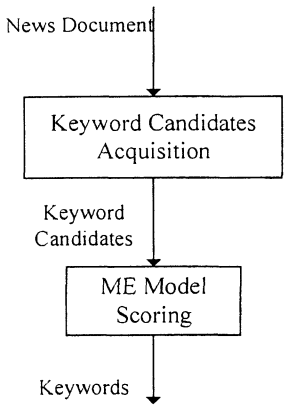

Figure 1: System architecture

The whole procedure is simple, mainly divided into two modules. Firstly, we need acquire keyword candidates from the news documents. Thus, it's must be decided what are possible keywords and how to get them. Secondly, all the keyword candidates are inputted into the maximum entropy model and scored according to their features.

Combined with the task of keyword indexing, we review the maximum entropy model. Here, the model is defined over X x Y, where X is the set of features which the candidates own, and Y is the indexing result set of {YES, NO}, and YES means that a candidate item belongs to a keyword, and NO means the opposite. ME model's probability of a feature set x together with the indexing result y satisfies exponential distribution, and has the form as formula (1).

$$p(y \mid x) = Z(x)\exp(\sum_i \lambda_i f_i(x,y))$$

$$Z(x) = 1/\sum_y \exp(\sum_i \lambda_i f_i(x,y)) \qquad (1)$$

where Z(x) is a normalization constant in a given context, $\{f_1, f_2, \cdots, f_n\}$ are known as features which are binary valued functions, and $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ are their corresponding parameters which indicate how important a feature is contributed to the model.

Given a sequence of candidate keywords with feature set $\{f_1, f_2, \cdots, f_n\}$ and $\{y_1, y_2, \cdots, y_n\}$ as training data, define F ($f_i \in F$) as the feature set available when predicting y_i. F includes features such as frequency, length, POS tags or phrase category, position and so on, which will be introduced in detail later. The parameters $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ are chosen to maximize the entropy of a distribution subject to certain constraints on feature set $\{f_i\}$. It has been proven that p(y|x) has the unique form which maximizes the entropy H(p). The model parameters for the distribution p(y|x) can be obtained via the Generalized Iterative Scaling (GIS) algorithm (Darroch et.al 1972). The normalization constant Z(x) is global and a single number Z(x) is used throughout the model.

During testing, we predict keywords for a new document. Firstly, candidates are generated automatically with the method introduced in section 3. Secondly for every candidate, we suppose that it will be a keyword and get a feature vector for it. According to formula 1, a probability of the candidate being keyword is calculated. Thirdly, we suppose that the candidate will not be a keyword and get another feature vector. Then, a probability of the candidate being not keyword will also be calculated. Lastly, the second probability divide the first probability, we can get a score, which is calculated as formula (2).

$$score(ck) = \frac{p(y = YES \mid x)}{p(y = NO \mid x)} \qquad (2)$$

Where ck means a candidate keyword, p(y=YES|x) means the probability that ck is supposed to be a keyword, whereas p(y=NO|x) means that ck is supposed not to be a keyword.

According to the length and content of a news document, we set the number of keywords to N. Then, the candidates that owe the N highest of the scores will be selected as keywords of the document.

3 Acquisition of Candidate Keywords

A document can be seen as a sequence of Chinese characters. If we extract all the possible character strings as candidates, the number is striking and it's not necessary to make such great effort. Then we should extract candidates according to the characteristics of news documents.

Firstly, a news document is always brief, and usually, only important terms repeat. Secondly, as a rule, the purpose of news documents is to illustrate an event or a thing for readers. Then this kind of documents usually place more emphasis on some named 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.

Aiming at the first characteristic of news documents, those terms with high frequency usually represent the main content of a document. Here, we don't use a dictionary, but get those terms only according to their frequency statistics (Liu 1998). We set a threshold value as 2 for the terms considering the length of news documents. Suppose that a character string is $c_1c_2 \quad c_n$, and $f(c_1c_2 \quad c_n)$ represents its frequency, then we extract $c_1c_2 \quad c_n$ as a term from text only if $f(c_1c_2 \quad c_n)$ equals to or more than 2. These candidates stand out just because of simple repetition and some of them are probably not meaningful units of language. We need to filter out those meaningless items, which you can refer to Li (2003) and isn't repeated here.

Due to the second characteristic of news documents, it is necessary to extract those named entities which occur only once. There are two kinds of named entities. The first are those which have rules of composition, mainly names and foreign terms. They can be recognized with statistical and rule-based methods combined, which we don't introduce here in detail. 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 hardly 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 the lexicon.

Those two kinds of candidate keywords can be obtained automatically. Every candidate is a keyword or it's not. We have concluded from about 100 news documents with keywords indexed by experts that about 96% keywords are included in the candidate set extracted through those two methods above. Next, it's better to accurately select only about 14% from every candidate set as keywords for every document on average. We can see the example data in table 1.

	National News(44)	International News(36)	Sports news (16)
By frequency statistics	172	114	54
Named Entity Recognition	40	45	27
Genuine keywords	220	162	87
Percent of extraction	96.4%	98.1%	93.1%
All Candidates	1636	1125	549
Percentage of	13.0%	14.1%	14.8%

Table 1: statistical data of candidates and keywords

4 Feature Representation

The advantage of maximum entropy model is that it can effectively incorporate diverse and overlapping features (Adwait 1996). Then, a feature set is crucial to the model to reach a good result of indexing. Here, we'll discuss how to obtain a feature set and what features are included.

Except the characteristics introduced above, there are some other characteristics which help us in keyword indexing. Contents with specific punctuations such as quotes (" ") and brackets (《》) are usually selected as keywords. In addition, the style of documents, part of speech, length and other characteristics of terms will also determine whether a term will be a keyword.

As in formula 1, the conditional probability $P(y|x)$ of a set of features x and indexing result y is determined by parameters of those features in the model, such as feature function $f_i(x,y)=1$. All features are selected by the model according to any appropriate factor that affects the indexing process. According to the characteristics introduced above, this feature set can include length, frequency, position, context (such as whether included in specific punctuations), part of speech or phrasal category and so on. And every feature must encode a binary value that might help predicting the indexing result.

In our model, we build a feature set in two steps. The first step is to build a set of feature templates according to the characteristics of news documents; and the second step is to obtain a set of concrete

binary valued feature functions by instantiating all the feature templates. Here, we define feature templates and their corresponding value range as in table 2.

No.	Meaning	Templates	Value Range
1	Length	LEN	Int{2~10}, MORE{>=10}
2	Frequency	FREQ	Int{1-19}, MORE{>=20}
3	Part of Speech or Phrase category	SYNTAG	{ NP, VP, OTHER}
4	Whether in special punctuations	IN_PUNC	{Quotes(" "), Brackets(《》), NONE}
5	Position	POSITION	{Title, First Paragraph, Last Paragraph, OTHER}
6	Document Style	STYLE	{National News, Sport International news, OTHER}
7	What type of named entities	NE_TYPE	{PERSON, PLACE, COUNTRY, ORGANIZATION,OTHER}
8	Indexing Result	DEFAULT	{YES, NO}

Table 2: Feature Templates and Value Range

Template 8 DEFAULT is a special template. Other templates must be incorporated with this template and assigned values in their value ranges. This can be seen the process of instantiating features. For example, as for the template LEN, we can assign it an integer ranged from 2 to 10. If the length is larger than 10, the template will be assigned the value MORE. Then according to the indexing result of the current candidate, the template DEFAULT will also be assigned a value. If the candidate belongs to a genuine keyword, its value will be YES, otherwise NO. Then these two instantiated templates are incorporated. So the other templates (Template 2~7) can do the same instantiation process. Here, the feature "LEN_2=YES" is formulated a binary value function as follows,

$$f_i(x,y)=\begin{cases}1 \; if \; \mathrm{y}=YES \text{ and } \mathrm{Len(x)}=2\\ 0 \; otherwise\end{cases}$$

The feature means that the length of the current candidate is composed of two Chinese characters, and at the same time, it is indexed as a keyword.

According to the feature templates and their value ranges, about 98 features are generated. Here we suppose that they are all useful to keyword indexing and don't need conduct the operation of feature induction. Then we use GIS algorithm to estimate parameters of all existing features in the model.

5 Experimental Results

We extracted about 2,260 candidate items automatically from about 100 news articles as training data, which include 2,015 negative cases and 245 positive cases. Negative means that a candidate isn't a keyword, and positive cases are keywords.

We have compared the results of ME model with the method of scoring by experience (Li 2003) and Chien's(1997), as in table 3. The performance is measured with precision and recall. Precision is the percentage of candidates that belong to keywords found by our system. Recall is the percentage of keywords found by the system present in the corpus. The recall of ME model is 0.05 lower than the method by experience, but higher than Chien's, and its precision is lower than those two methods.

	Recall			Precision		
	Experience	ME	Chien's	Experience	ME	Chien's
National politics (23)	0.452	0.405		0.401	0.359	
International Politics (10)	0.644	0.593		0.594	0.547	
Sports news (4)	0.629	0.568		0.482	0.435	
Average	0.523	0.473	0.30	0.462	0.418	0.43

Table 3. Experimental Results

The result is not very satisfying. However, we think that this method has potential power. We know that ME model need a large collection of training data to evaluate feature parameters well. Due to the limit of training corpus, there are only about 2,000 cases and 100 features available for the ME model, but the final results almost reach the state-of-the-art accuracy. Thus, we can conclude that the fault doesn't lie in the ME model itself. With training data increasing, the result will improve.

6 Conclusions

We have described a system for automatically indexing keywords from texts based on ME model. Here we utilize the mature techniques available now such as string frequency statistics, segmentation and POS tagging tools to obtain candidate keywords. Then, according to features, we propose ME model to evaluate directly every candidate keyword and select those with higher scores as keywords. ME model itself offers a clean way to conduct keyword indexing. It can use diverse pieces of linguistic evidence to predict the probability distribution of two possible indexing results for every candidate items. The experimental results show that our system can perform comparably to the state of the art.

Owing to the limits of the training corpus, the parameters of features in ME model are evaluated, but not very exactly. Then, we need cumulate more and more documents with keywords. Then we can make parameters more objective. That will be our further work.

Acknowledgements

Our thanks go to Ye Jiaming, Zhu Anfeng, Geng Xiangyu and all other colleagues of ICL for their help on this work.

References

Adwait Ratnaparkhi, 1996, A maximum entropy model for part-of-speech tagging, In Proceeding of the Conference on Empirical Methods in Natural Language Processing.

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

Darroch, J.N. and Ratcliff, D. 1972, Generalized Iterative Scaling for Log-Linear models, Annals of Mathematical Statistics, 43(5): pp. 1470-1480.

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

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

Sujian Li, Houfeng Wang, Shiwen Yu and Chengsheng Xin, 2003, News-Oriented Automatic Chinese Keyword Indexing, Sighan Workshop, Sapporo, Japan.

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

Wenfeng Yang, 2002, 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, pp. 439-440.

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