Automatic Domain-specific Term Extraction System Based on Hybrid Approaches

Miao Wan, Song Liu, Cong Wang

Center for Intelligence Science and Technology Research, Beijing University of Posts and Telecommunications, 100876

E-mail: wanmiao120@163.com

Abstract: In this paper we establish a multi-strategy-based automatic Chinese term extracting system combining both statistics-based and rule-based methods. A new metric named term association (TA) in the statistical part is proposed to measure the combining degree between two character strings, and we get a candidate list in this part. Then domain-specific terms are selected based on defined rules. This system takes unprocessed technical papers of "ethanol fuels" as input. The results of experiments are analyzed and evaluated, and a higher precision than the previous approaches is achieved.

1 Introduction

Term extraction from Chinese texts is quite difficult especially for unknown words and compound key words, such as names, locations, translated terms, technical terms, abbreviations etc [1, 2]. Unlike English, Chinese language does not have explicit word boundaries in written sentences. How to detect term automatically has been a critical problem in Chinese information retrieval (IR) and Chinese natural language processing (NLP). Therefore, researchers in many countries have explored a number of new algorithms and techniques to solve the problems of automatic term acquisition, and there are three main methods for term detection at present: statistical, rule-based and hybrid approaches.

Statistical approaches gather statistical features such as the frequency of words and their co-occurrence. Previous researchers focus on the use of one or two of these features [3]. For example, mutual information (MI) is used to measure the association of character strings which compose a term or a candidate [4, 5]. Another feature in using is the likelihood score [6] that represents, given a word as well as its part-of-speech tag and length, how likely a character appears in certain position within the word. Other features such as in-word probability [7], context dependency [8], and relative frequency [9] are also used in term detection. They are almost domain-independent, although they require a large amount of training data, which should sometimes be prepared manually.

Rule-based approaches detect unknown keywords by using a dictionary and heuristic rules for forming words. These type of methods need to make amount of templates for complex term structures, so that they are not easily adapted to other domains. It is neither possible for a dictionary to contain all the words in Chinese nor to specify all the rules for word formation. However, for a domain-specific term extraction task, rule-based approaches are quite effective in a certain degree, which is a great help in our research work.

The hybrid approaches have been applied in English term extraction systems already and get well results. But for Chinese, most automatic term selection methods are statistics-based. Du (2005) [10] designed a multi-strategy based term extracting algorithm with a hybrid method and improved the statistical algorithms.

Agreeing with Du, we introduced a new hybrid approach which combined the statistical and rule-based methods together. During the statistics part, we proposed a compound word extraction algorithm based on a new statistical metric called Term Association (TA) which measured the

* The research work in this paper is supported by the research Fund for PhD students, Ministry of Education (20060013007), Beijing Municipal Natural Science Foundation (4073037).

combined degree of two neighbor strings. And then in the second part of our system, we select the highly domain-specific compound words to be technical terms by defined rules. Our purpose is to achieve a higher precision of the domain-specific Chinese term extraction system by the hybrid method than the previous approaches.

The approach proposed in this paper is a four-step process. Each step can be seen as a dependent module. The corresponding flowchart of this automatic process is given by Figure.1.

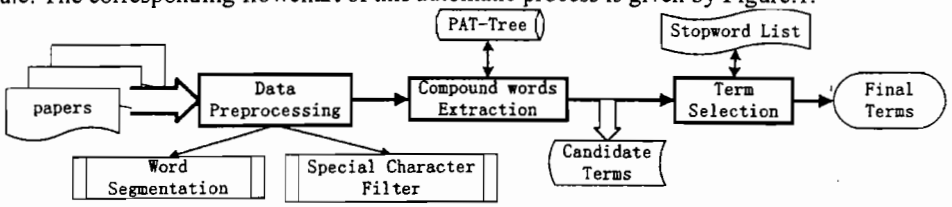

Figure 1. The process of our term extraction approach

2 Corpus Storage in PAT-tree Structure

Since the traditional N-gram language model is not a meaningful unit in linguistics, N-gram-based indexing tends to cause inconsistencies between training data and testing data when a term appears in the training data but not in the testing data [11]. So for the efficient search and convenient string update, before the extraction task, we built PAT-tree as the data structure for the corpus storage.

PAT-tree was developed by Gonnet [12] from Morrison's PATRICIA algorithm (Practical Algorithm to Retrieve Information Coded in Alphanumeric) [13]. The PAT-tree is conceptually equivalent to compressed digital search tree but smaller. The superior feature of the PAT-tree data structure is most resulted from the use of so-called semi-infinite strings [14] in storing the substring values in the nodes of the PAT-tree. Distinguished with N-gram language model, the searched strings in the PAT-tree do not restricted by the length. Using this data structure to index the full-text of documents, all possible character strings, including their frequencies in the documents, can be retrieved and updated in a very efficient way, yet not every character string with arbitrary length is needed to be stored.

At present, this vary-gram language model is successfully used in the area of IR. There is an example in Figure. 2 which shows the semi-infinite strings generated for the Chinese character sting "中国汽车领域石油替代可能性" and its storage in the PAT-tree.

Data string:

1	2	3	4	5
中国	汽车领域	石油	替代	可能性

PAT-tree:

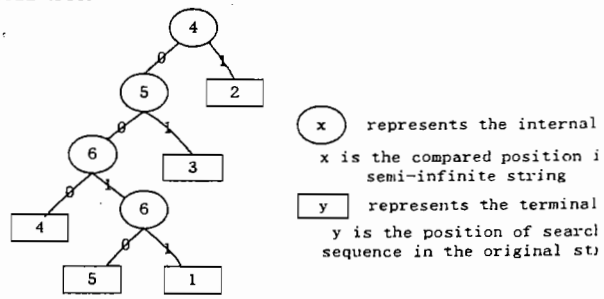

Figure 2. An example of the Chinese PAT-tree

Basically, although PAT-tree is a very efficient data structure to record all of the substrings of documents, it really demands large space overhead and takes time to build. In Section 4 and 5, some empirical results on the construction of PAT-tree will be reported.

3 Automatic Term Extraction Approaches

3.1 Data Preprocessing

The input of our system is unprocessed technical papers about "ethanol fuels". At the beginning of our approach, word segmentation by using the backward-maximum-method and part of speech (POS) tagging was carried out on the collected texts. And for higher precision of our method, we filtered a few special characters and symbols that are rarely used.

3.2 Compound Word Extraction Based on Statistical Methods

Because technical terms are mostly compound words, the results of word segmentation should be connected to compound words. After data preprocessing for the corpus, the further analysis is to choose reasonable statistical features to detect unknown compound words heuristically. Although the features proposed in Section 1 are all valuable for term extraction, only one feature used separately in the approach may be not quite satisfied in the value precision and recall. In order to make our approach more feasible, we proposed combining two complement features (MI and Log-Likelihood Ratio) into a new metric called Term Association (TA) and explore an estimation function to acquire unknown compound words. Keywords that given by each paper with these detected compound words will form a term candidate list after this part. These candidates are ultimately validated as entries of term selection in Section 4. All the statistical and linguistic information such as word's frequency and character word number of corpus can be easily obtained from the PAT-tree we have just established.

3.1.1 Mutual Information(MI) and Log-Likelihood Ratio (logL)

For any two neighbor strings (x and y), the Mutual Information of them is defined as follows:

$$MI(x,y) = \log \frac{P(x,y)}{P(x)P(y)} \qquad (1),$$

where $P(x,y) = \frac{C(x,y)}{C(*)}$, $P(x(y)) = \frac{C(x(y))}{C(*)}$, $C(x)$ is the occurrence number of the variable x as $C(x, y)$ is the occurrence number of the variable xy. $C(*)$ represents the total number of words in the corpus.

Theoretically, mutual information measures the information that x and y share: it measures how much knowing one of these variables reduces our uncertainty about the other. The larger MI of xy's co-occurrence, the more chance it has to be a term candidate.

For x and y, their log value of likelihood score (logL) is defined as follows:

$$\log L(x,y) = l(\frac{k_1}{n_1}, k_1, n_1) + l(\frac{k_2}{n_2}, k_2, n_2) - l(\frac{k_1+k_2}{n_1+n_2}, k_1, n_1) - l(\frac{k_1+k_2}{n_1+n_2}, k_2, n_2) \qquad (2),$$

where $l(p,k,n) = k\log(p) + (n-k)\log(1-p)$, and $k_1 = C(x,y)$, $n_1 = C(x,*)$, $k_2 = C(\neg x, y)$, $n_2 = C(\neg x, *)$

Mostly, $C(x,*)$ approximately equals with $C(x)$, so in this paper we use $n_1 = C(x,*) \approx C(x)$, $k_2 = C(\neg x, y) = C(*, y) - C(x, y) \approx C(y) - k_1$, $n_2 = C(\neg x, *) = C(*) - C(x, *) = C(*) - n_1$ in our calculation.

3.1.2 Term Association(TA)

We have already observed that ranking the candidate terms merely by the two statistic measures mentioned before are not quite satisfied. So we use a new statistic called "Term Association" (TA) that combined the two explored features into a new statistical estimation function to measure the strength of association between two

neighbored character strings. For any two neighbored strings x and y in the corpus, their TA(x, y) can be calculated as follows:

$$TA(x,y)=\begin{cases} 0, & C(x,y)<cThread \\ 0, & C(x,y)\geq cThread \ and \ \log L\leq lThread \\ \frac{MI}{2}+\frac{\log L}{2}, & C(x,y)\geq cThread \ and \ \log L>lThread \end{cases} \quad (3),$$

where *cThread* and *lThread* are two constant threshold values chosen by experience.

With a threshold *tThread* of TA, we can determine whether two words can be connected as a new candidate. If $^{TA(x,y)}>tThread$, x and y will form a new phrase xy as a candidate term.

The whole procedure of compound words extraction obeyed the forward-maximum-combination rule. The calculation process can be express as:

Step 1: Read one line of a target document.

Step 2: If this line is not empty, then split it with the symbol of space.

Step 3: For the first two neighbor strings of the line, take TA calculation for them. If their TA value is lager than the value of *tThread*, they will be connected together. If not, go for the next two strings.

Step 4: Repeat **Step 3** until the end of the line.

Step 5: Repeat from **Step 1** to **Step 4** until the end of this document.

3.3 Term Extraction Based on Defined Rules

For the academic paper, not all the acquired candidates are the target terms after the above extraction algorithm. In some cases it may not work only based on the statistical metric TA. Some compound candidates that have large TA values are not terms in the specific field. For instance, the new phrase "辛烷值高" is a candidate but not a term. Since the composed words "辛烷值" and "高" co-occurred frequently that their TA value is larger than *tThread*.

While previous researches show that most domain-specific terms have some linguistic and statistical features and they do help in term detecting. For instance, in technical documents, the majority of domain-specific terms are noun phrases or compound nouns consisting of a small number of single nouns. We then investigate several common patterns that can be used in term selection to improve the performance of our approach. In this case, we define four rules to filter unwanted candidates from the candidate list. With these rules and a well prepared stop-word list, our system can determine whether a character string is a term or not automatically.

For easy understanding, these rules can be explained as follows:

For $\forall c \in CList$,

(1) if c is not composed as the lexical pattern of {*((A|N)*(NP)*(A|N)*)N*}, then $c \notin TList$;

(2) if c is composed as *"xsy"*, and $x \in STList$ *or* $y \in STList$, then $c \notin TList$;

(3) if $C(c)>hThread$, then c is not a term;

(4) if $D(c)<dThread$, then c is not a term.

Here A is representation of adjective, N of noun and NP of noun phrase. (.)* in the rules represents that the pattern in the parentheses appears at least once. $D(x)$ represents the document frequency (df) of x and the *CList*, *TList* and *STlist* are representation of candidate list, term list and stop-word list respectively.

4 Experiments

To get an elaborate evaluation for how well our approach performs and for the further research, we developed a platform to implement the term extraction system of technical domain. The experiments on this platform were done to extract terms from a set of technical papers in the field of ethanol fuels, and we made a front-interface for this system by JSP so that we can see the results of each step clearly. Figure 3 below show the index page of our system:

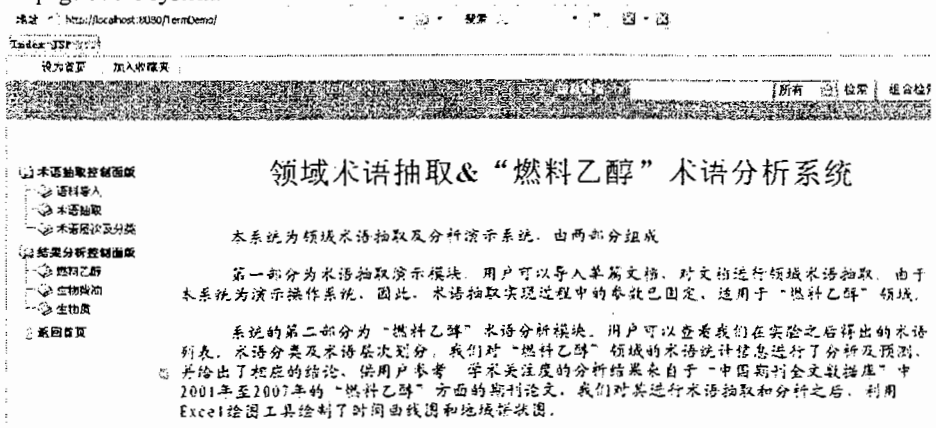

Figure 3. The index page of our term extraction system

The papers had been indexed manually first and a total of 285 terms extracted. Then we did three parallel experiments to compare our approach with the statistic-based method and Du's extractor. We selected most of the parameters by experience. The chosen thresholds are shown in Table 1.

Table 1. Thresholds chosen in the experiments

cThread	*lThread*	*tThread*	*hThread*	*dThread*
3	20	30	15	5

In statistics-based extractor, only the first three thresholds were selected. And in Du's extractor, only the first two parameters and *hThread* are used. All the application procedures were executed by Java programs automatically, and Table 2 shows the space and speed performance of our extractor ran on Java platform.

Table 2. The performance obtained in the experiments for the term extraction application

Text size	1.93(MB)		Time of selecting final terms	1889(ms)
PAT-tree size	5.20(MB)		Number of extracted candidate terms	430
Time of building PAT tree	5366610 (ms)		Number of selected final terms	215
Time of extracting candidate terms	7382938(ms)		Number of documents in the corpus	183

After the whole processes, we got the final terms. The displayed results can be shown in Figure 4 below:

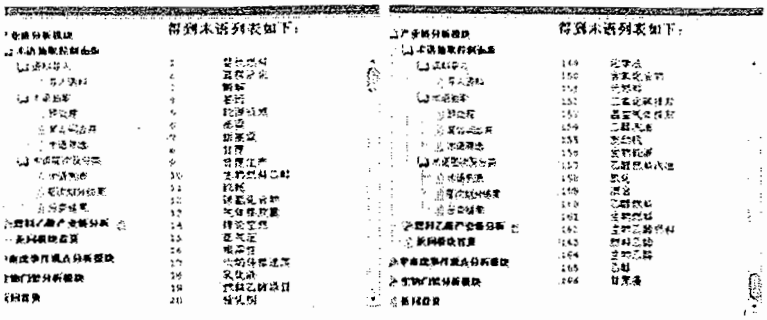

Figure 4. The final terms displayed on the front-interface after the extraction task

5 Conclusion

5.1 Evaluation

In this study manual work has been carried out in order to realize the performance. We got a term list as the target documents have been manually extracted. We made a program to estimate whether the result term we got after the experiments and the one in the correct term list are match exactly or not.

The statistic-based extractor can be seen as a sub-module of our extraction system, so the candidate terms we got can be taken as its final terms. Du's extractor combined MI and logL in a different way from us, and in the term selection part, it chose only one parameter to filter the candidates. Table 3 gives the results of our evaluation task which compared with the performance of statistical method and Du's term extractor.

Table 3. Some results obtained from the experiments

	Statistic-based Extractor	Du's Extractor	Our Extractor
Number of Selected Terms	430	180	216
Number of Correct Terms Extracted	177	116	182
Obtained Precision	0.4116	0.6449	**0.8426**
Obtained Recall	0.6211	0.4064	**0.6386**

5.2 Results

Under the help of the TA-based mixed approach, the application of technical paper term extraction got 216terms with the precision of 84.26% and the recall of 63.86%. The precision is lager than the statistics-based methods (60.47%) and this is a significant improvement over the precision of 64.49% and the recall of 40.64% given by Du's extractor [10]. For the recall issue, we obtained 182 correct terms, which are more than the other two extractors.

6 Further Study

The proposed PAT-tree does reduce the difficulty of term extraction in Chinese-included documents, which is fundamental to our task. Meanwhile, it takes much time on PAT-tree building and it is a problem for high efficiency of our system. In the next step of our study, we will try to improve its time performance.

Term extraction can be used in many fields of NLP and IR. It is also valuable in concept acquisition during the knowledge base construction. We have planned to cluster terms after the extraction task, divide them into certain categories and discover the deeper information from these domain-specific terms.

References

[1] Keh-Jiann Chen et al., "Word Identification for Mandarin Chinese Sentences", COLING'92.

[2] Lee-Feng Chien. "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval," Proceedings of SIGIR'97, 1997, pp. 50-58.

[3] Hongqiao Li, Chang-Ning Huang, Jian-feng Gao and Xiao-zhong Fan, "The Use of SVM for Chinese New Word Identification". In IJCNLP-04. Sanya City, Hainan Island, China, March 22-24, 2004.

[4] Keh-Yih Su, Ming-Wen Wu, and Jing-Shin Chang, "A Corpus-based Approach to Automatic Compound Extraction", In Proceedings of ACL 94, 1994, pp. 242-247.

[5] Jian Zhang, Jian-feng Gao, and Ming Zhou, "Extraction of Chinese Compound Words: An Experimental Study on a Very Large Corpus", Proceedings of the Second Chinese Language Processing Workshop, 2000, pp. 132-139.

[6] Andi Wu. "Chinese Word Segmentation in MSRNLP". In proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, July 11-12, 2003.

[7] Aitao Chen. "Chinese Word Segmentation Using Minimal Linguistic Knowledge". In proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, July 11-12, 2003.

[8] Sheng-fen Luo, Mao-song Sun. "Two-Character Chinese Word Extraction Based on Hybrid of Internal and Contextual Measures". In proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, July 11-12, 2003.

[9] T. H. Chiang, Y. C. Lin and K.Y. Su. "Statistical models for word segmentation and unknown word resolution". In proceedings of the 1992 R. O. C. Computational Linguistics Conference, Taiwan, 1992, pp. 121-146.

[10] Du B, Tian HF, Wang L, Lu RZ. "Design of domain-specific term extractor based on multi-strategy". Computer Engineering, 2005, 31 (14), pp.159–160.

[11] Yu-Sheng Lai and Chung-Hsien Wu, "Meaningful Term Extraction and Discriminative Term Selection in Text Categorization via Unknown-Word Methodology", ACM Transactions on Asian Language Information Processing, Vol. 1, No. 1, March 2002, pp.34-63.

[12] Gaston H. Gonnet, Ricardo A. Baeza-yates and Tim Snider, "New Indices for Text: Pat Trees and Pat Arrays", Information Retrieval Data Structures & Algorithms, Prentice Hall, 1992, pp. 66-82.

[13] Morrison, D., "PATRICIA: Practical Algorithm to Retrieve Information Coded in Alphanumeric", JACM, 1968, pp. 514-534.

[14] Manber, U. and R. Baeza-Yates, "An Algorithm for String Matching with a Sequence of Don't Cares", Information Processing Letters, 37, 1991, pp. 133-136.

Appendix A: A part of stop-word list

我国 稍微 少许 例如 绝对 开始 几乎 接近 必然 关于 各种 否则 年代 包括 别 并 更 凡 当 便 被 啊 呢 哦 噢 把 吧 而 得 吨

Appendix B: All terms extracted from the technical paper.

替代燃料 直接液化 酶解 基因 能源领域 高粱 甜高粱 甘蔗 甘蔗生产 生物燃料乙醇 能耗 碳氢化合物 气体排放量 理论空燃 蒸气压 相溶性 火焰传播速度 乳化液 燃料乙醇项目 催化剂 生物燃料产业 纤维素乙醇 玉米 玉米乙醇 纤维乙醇 燒值 排放特性 裂解 相溶 助溶 能源替代 添加量 空燃比 蒸发潜热 溶胀 厌氧 液体燃料 乙醇含量 理化特性 直接乙醇燃料电池 转化效率 含氧量 玉米加工 可再生燃料 直接乙醇燃料 化石燃料 纤维质 纤维素酶 乙醇发动机 乙醇氧化 平均有效压力 混合动力汽车 能源作物 热效率 乙醇燃料均质压燃 平均指示压力 含水乙醇 能源植物 矿物燃料 植物纤维 乙醇电氧化 乙醇电催化氧化 乙醇生产 汽油辛烷值 负荷特性 滞燃 滞燃期 燃烧持续期 高热效率 进气预热 发酵 火花点火 爆震边界 混合气要 空气系数 强吸附 爆震限制线 排放影响 乙醇燃料发展 掺混 热值 负荷工况 空燃 燃油 含氧燃料 压燃 深加工 化学反应动力学 进气温度 燃烧程 活化 内压力 抗爆性 灵活燃料 普通汽油 车用乙醇汽油 汽油机 生物酶 乙醇产量 均质压燃 轻型汽车技术 混合动力车 乙醇燃料车 热解 气化 燃料 替代 非粮乙醇 燃料乙醇生产 掺入 生物资源 农产品 再生能源 理化性质 理论空燃比 汽化潜热 着火极限 混合气热值 石油燃料 抗爆 碳烟排放 无水酒精 变性燃料乙醇 互溶 助溶剂 氢燃料 碳排 酶技术 液化 催化 燃料电池 排气 抗爆性能 经济性 汽油 能量密度 辛烷值 汽车燃料 生产乙醇 替代能源 粮食乙醇 陈化粮 掺烧 醇燃料 二氧化碳 无水乙醇 生物技术 可再生能源 化学法 含氧化合物 代燃料 二氧化碳排放 温室气体排放 乙醇汽油 发动机 生物能源 乙醇燃料汽油 氢化 混合乙醇燃料 生物燃料 生物乙醇燃料 燃料乙醇 生物乙醇 乙醇 甘蔗渣