A High Precision Algorithm for Automatic Extraction of High-frequency Words Based on Statistics

XUAN Zhaoguo, DANG Yanzhong, JIANG Shaohua, ZHAO Mingwei
Institute of Systems Engineering, Dalian University of Technology, Dalian, 116024, China
xzg@dl.cn

ABSTRACT

Automatic Chinese Word Segmentation is one of the basic research issues on text categorization, automatic summarization and information retrieval as well as other Chinese Information Processing tasks. In this paper we put forward a high precision algorithm for extracting high-frequency words without thesaurus. It firstly counts the frequencies of co-occurrence patterns of Chinese characters from documents, then eliminates the "bridge-connection" frequencies and therefore obtains the support frequencies of patterns. Afterwards, the words are identified and acquired according to the support frequencies instead of the primary appearing frequencies. The proposed algorithm is tested in the task of extracting words from several sets of scientific document abstracts, and the results show that this algorithm can improve both precision and recall of extracted lexical set to some extent. This algorithm can either be applied to text categorization and automatic summarization.

Keywords: Chinese-word segmentation, statistics algorithm, high-frequency words, Chinese information processing

1. INTRODUCTION

Automatic Chinese Word Segmentation (ACWS) is one of the basic research issues on Chinese Information Processing (CIP) tasks such as text categorization, automatic text summarization, information retrieval, natural language understanding, and so on[1]. ACWS has become a hot topic since the results of word segmentation affect directly this CIP tasks. Up to date, methods of word segmentation can be roughly classified into two categories, namely thesaurus based methods and statistics based methods.

Now in ACWS, the mainstream algorithms are based on thesaurus, and the technologies in which are quietly mature. But, the words included in a thesaurus are limited and cannot adapt to new application domains. Such algorithms are not able to handle the words not existing in the thesaurus, especially for important high-frequency words

Therefore, some researchers focus on the segmentation technologies without thesaurus[2-5]. A word is a co-occurrence pattern of Chinese characters, but not every pattern can be of the word. Only those with the real semantic meanings or grammatical functions are regarded as words. Without thesaurus, Chinese words cannot be recognized because of lacking of a priori knowledge. But from the viewpoint of statistics, the patterns of Chinese characters in each document are observable. For most documents, it is very easy for a word to be recognized if it appears many times. On the contrary, it is difficult to find a word if it appears few times in the documents. The main idea of these kinds of methods is to judge whether a co-occurrence pattern of Chinese characters is a lexical word or not by its frequency. But some patterns are meaningless even they have high frequencies. How to filter out these meaningless patterns is still a problem.

In this paper we put forward a high precision algorithm for extracting high-frequency words without thesaurus. It firstly counts the appearing frequencies of co-occurrence patterns of Chinese characters from documents, then eliminates the "bridge-connection" frequencies by the iterative algorithm, and extracts the words according to the modified frequencies instead of the original appearing frequencies. Experimental results show that the proposed algorithm can filter out some meaningless patterns and therefore improve the precision of extracted lexical set.

The rest of this paper is organized as follows: In Section 2, we introduce the widely used model for extracting words based on statistics. Our improved algorithm is proposed in Section 3. Experimental results are reported in Section 4. And we give the conclusion in Section 5.

2. WORD EXTRACTING MODEL

The procedure of extracting words includes 3 main steps: preprocessing documents, counting the frequencies of patterns and filtering patterns, as shown in Fig. 1 below.

* This paper is based on work supported by the National Natural Science Foundation of China under project 70271046

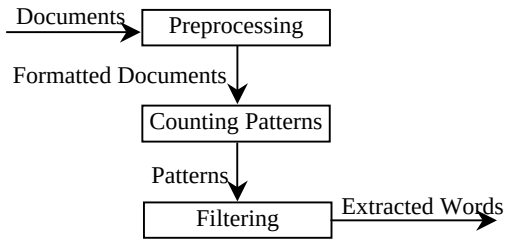

Figure 1. Steps for Word Extraction

During the preprocessing of documents, the character that cannot be used to form a word, such as interpunction, number and the other non-Chinese character, is replaced by a special symbol which is called the space-character. After preprocessing, the documents are transformed to a string of characters, $s=s_1s_2\ldots s_n$, $s_i \in H \cup \{\square\}$. Here H denotes the corpora of Chinese characters, and $\square$ denotes the space-character.

The second step is to count the frequency of each pattern. The patterns of Chinese characters are usually captured by N-gram. Once the minimum length (M) and maximum length (N) of words are predetermined respectively, we can get a set of n-grams (n=M…N). By excluding the patterns containing "□", we get all of the patterns appearing in documents, e.g. for string s="*abcde□bcd*" if M=2 and N=4, 9 patterns can be gotten, they are "*ab*", "*abc*", "*abcd*", "*bc*", "*bcd*", "*bcde*", "*cd*", "*cde*", and "*de*". Then we can count the appearing frequency of each pattern, denoted by $\Gamma(\omega, s)$. For the above example, Γ("*bc*",s) = Γ("*bcd*",s) = Γ("*cd*",s) = 2.

The third step is to filter the patterns. The following methods are concerned with this problem. HAN Kesong[2] and HE Hao[6] selected the patterns with high frequencies via setting an appropriate threshold. JIN Xiangyu[3] analyzed the inclusion relationship of strings, and introduced some criteria such as support and confidence to filter these co-occurrence patterns to get lexical items. Some other researchers integrated frequencies and rules or knowledge to choose lexical features, e.g. WU Yingliang[4] integrated N-gram models and machine learning.

By analyzing the results obtained from the N-gram algorithm, we find an erroneous pattern that is called "bridge-connection" pattern in this paper. In such situation, the original appearing frequency of the pattern cannot be used for the judgment for the pattern being a word. In Section 3 we will discuss the process where the "bridge-connection" patterns form and the way to filter out these kinds of patterns in detail.

3. "BRIDGE-CONNECTION" PATTERNS FILTERING

3.1. "Bridge-connection" Pattern and Its Support Frequency

For expressing clearly, we give some definitions at first.

The string $s_i\ldots s_j$ in the given document is denoted by $s_{<i,j>}$. The pattern of Chinese character is denoted by ω. The set of all patterns in the document is called the set of candidate words, denoted by ψ. The set of recognized words denoted by Ω is selected from the candidate words.

Definition 1：For $\omega_1 \in \psi$ and $\omega_2 \in \psi$, if ω_2 is formed by ω_1 and a postfix string, *suff*, that is $\omega_2=\omega_1+suff$ (or by ω_1 and a prefix string, *pref*, that is $\omega_2=pref+\omega_1$), then ω_1 is called the sub-pattern of ω_2. ω_2 is called the extended pattern of ω_1. The set of all extended patterns of ω_1 is denoted by $Ex(\omega_1)$.

Definition 2： Suppose pattern $\omega_1=s_{11}s_{12}$, and pattern $\omega_2=s_{11}s_{12}s_{<1,k>}$ (or $\omega_2=s_{<1,k>}s_{11}s_{12}$), it is obvious $\omega_2 \in Ex(\omega_1)$. If ω_1 is recognized to be a word, the pattern $\omega'=s_{12}s_{<1,k>}$ (or $\omega'=s_{<1,k>}s_{11}$), is called a "bridge-connection" pattern. As ω' is a sub-pattern of ω_2, the frequency of ω' covers the frequency of ω_2. We call the frequencies derived from bridge-connection situation the bridge-connection frequency of ω', denoted by $\Gamma_B(\omega')$. The difference between $\Gamma(\omega', s)$ and $\Gamma_B(\omega')$ is called the support frequency of ω' and denoted by $\Gamma(\omega')$, that is $\Gamma(\omega') = \Gamma(\omega', s) - \Gamma_B(\omega')$.

The typical reason for generating such patterns lies in: for many pairs of words, if the anterior words end at the same character and the posterior words begin at another same character, the last character of the anterior words and the first character of the posterior words may form a co-occurrence pattern named "bridge-connection", which may have the high frequency. This erroneous pattern may have perfect support and confidence and hereby is incorrectly extracted as a word which results in a lower segmentation precision.

3.2. Patterns Filtering

A "bridge-connection" pattern may be formed by two adjacent words. It is comprised by the ending part of the anterior word and the beginning part of the posterior word. Most bridge-connection patterns are meaningless. Therefore, such patterns should be filtered out according to the predefined support frequencies instead of the

original appearing frequencies. Here we use support frequency of a pattern to redefine its support and confidence which are computed by the following formulas.

For $\omega^* \in \psi$, if there exists $\omega' \in Ex(\omega^*)$, and $\Gamma(\omega') = \underset{\omega \in Ex(\omega^*)}{Max}(\Gamma(\omega))$, the support and confidence are respectively defined as follows:

$$Supp(\omega^*) = I(\omega^*) - I(\omega') \quad (1)$$

$$Conf(\omega^*) = 1 - I(\omega')/I(\omega^*) \quad (2)$$

Specially, if $Ex(\omega^*)$ is null, then

$$Supp(\omega^*) = I(\omega^*) \quad (3)$$

$$Conf(\omega^*) = 1 \quad (4)$$

Support and confidence reflect the dependency of a pattern on a special context. If their values are small, the pattern may strongly relay on a special context and hence the probability of being a word is low.

Suppose two thresholds θ_1 and θ_2 be predefined by users. If $Supp(\omega) \geq \theta_1$ and $Conf(\omega) \geq \theta_2$, then pattern ω is considered to be a word.

3.3. Support Frequency Computing

In general, the expression of computing support frequency of $\omega = s_x s_y$ is :

$$I(s_x s_y) = I(s_x s_y, s) - I(s_a s_x s_y, s) - I(s_x s_y s_b, s) + I(s_a s_x s_y s_b, s) \quad (5)$$

Where $s_a s_x \in \Omega$, $s_y s_b \in \Omega$.

The iterative algorithm is presented below:

1. Set the thresholds of $Supp(\varphi_1)$ and $Conf(\varphi_2)$;
2. Ω := null;
3. foreach ($\omega \in \psi$)
4. $\quad I(\omega) := I(\omega, s)$
5. WHILE (true)
6. {
7. $\quad$ Search $\omega^* \in \psi$, and $\Gamma(\omega^*) = \underset{\omega \in \psi}{Max}(\Gamma(\omega))$, here $\omega^* = s_a s_b$,
8. $\quad$ IF $I(\omega^*) < \varphi_1$ THEN STOP;
9. $\quad$ IF $Supp(\omega^*) \geq \varphi_1$ and $Conf(\omega^*) \geq \varphi_2$ THEN
10. $\quad$ {
11. $\quad\quad \Omega := \Omega \cup \omega^*$
12. $\quad\quad$ foreach s_t, and $s_a s_b s_t \in Ex(\omega^*)$
13. $\quad\quad$ {
14. $\quad\quad\quad I(s_b s_t) := I(s_b s_t) - I(s_a s_b s_t, s)$
15. $\quad\quad\quad$ foreach s_x and $s_t s_x \in \Omega$
16. $\quad\quad\quad\quad I(s_b s_t) := I(s_b s_t) + I(s_a s_b s_t s_x, s)$
17. $\quad\quad$ }
18. $\quad\quad$ foreach s_t, and $s_t s_a s_b \in Ex(\omega^*)$
19. $\quad\quad$ {
20. $\quad\quad\quad I(s_t s_a) := I(s_t s_a) - I(s_t s_a s_b, s)$
21. $\quad\quad\quad$ foreach s_x, and $s_x s_t \in \Omega$
22. $\quad\quad\quad\quad I(s_t s_a) := I(s_t s_a) + I(s_x s_t s_a s_b, s)$
23. $\quad\quad$ }
24. $\quad$ }
25. }

The comparison of original appearing frequencies with support frequencies is shown in Table 1.

Table 1. The comparison of original appearing frequencies with support frequencies

No.	Pattern (ω)	$I(\omega, s)$	$I(\omega)$
1	服务	137	137
2	协议	113	113
3	商务	54	54
4	协作	32	32
5	**务协**	**26**	**0**
6	商务协	18	0
7	务协议	18	0
8	服务协	8	0
9	务协作	8	0

From the above table we can see that pattern ω_5, which is meaningless, will be considered to be a word according to the appearing frequency. However, it can be filtered out based on the supporting frequency using our iterative algorithm.

4. EXPERIMENTAL RESULTS

We select three groups with the total number of 100, 400, 1000 abstracts of scientific papers as the experimental data, and denote them by S_{100}, S_{400} and S_{1000} respectively. In the experiment, the confidence is set to be 0.1. The experimental results with different supports are shown in Table 2.

When $\varphi_1 = 4$ and $\varphi_2 = 0.1$, we rank each group of words with their corresponding supports in descending order and divide each group into 10 equal scales that construct a set of "recall-precision" testing points. The recall corresponding to i^{th} testing point is defined as $R_i = N_i/M$, where N_i denotes the number of the words extracted correctly in the first-i scales and M denotes the number of the words gotten manually. The precision corresponding to i^{th} testing point is defined as $P_i = N_i/T_i$, where T_i denotes the number of the words extracted in the first-i scales. The experimental results are shown in the following figures.

Table 2. The results with different support

Supp.	S_{100}			S_{400}			S_{1000}		
	Number of obtained words	Number of correct words	Prec.	Number of obtained words	Number of correct words	Prec.	Number of obtained words	Number of correct words	Prec.
2	850	768	90.35	1915	1550	80.94	3669	2616	71.30
3	808	731	90.47	1855	1501	80.92	3553	2551	71.80
4	594	568	95.62	1418	1262	89.00	2666	2184	81.92
5	477	463	97.06	1166	1082	92.80	2225	1931	86.79
6	402	394	98.01	1014	952	93.89	1964	1739	88.54
7	328	322	98.17	902	856	94.90	1751	1575	89.95
8	278	273	98.20	811	783	96.55	1555	1417	91.13
9	250	246	98.40	732	711	97.13	1417	1310	92.45
10	214	211	98.60	674	657	97.48	1313	1226	93.37

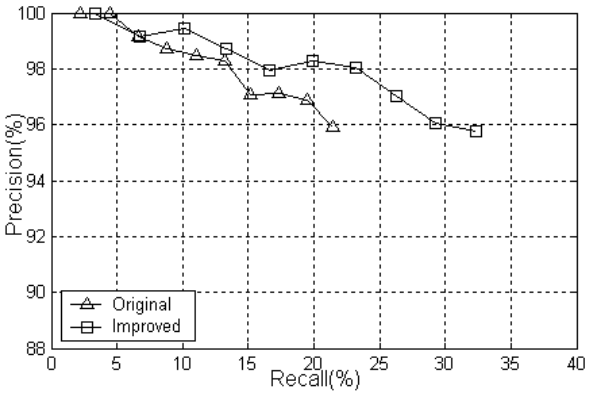

Figure 2. The Curve of "Recall-Precision" of S_{100}

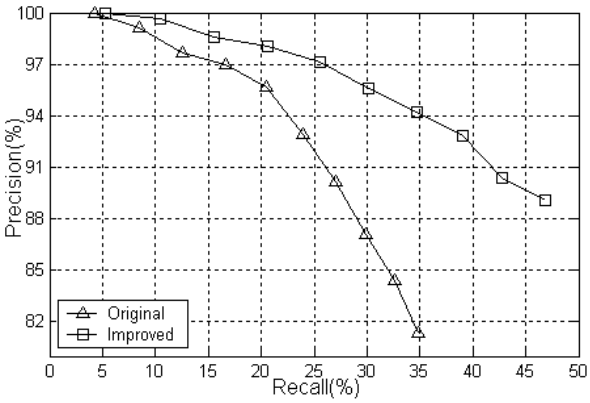

Figure 3. The Curve of "Recall-Precision" of S_{400}

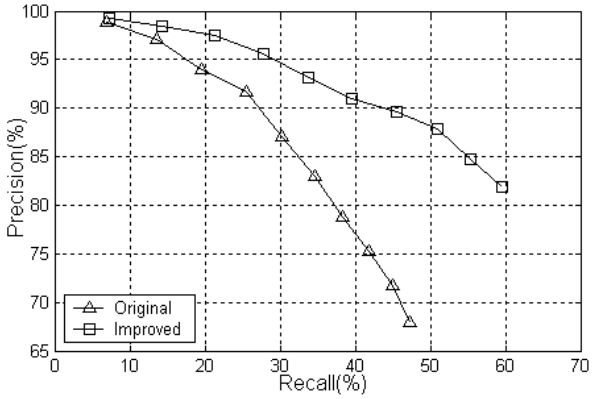

Figure 4. The Curve of "Recall-Precision" of S_{1000}

The results according to appearing frequencies are denoted by "Original", and those according to support frequencies are denoted by "Improved". The experimental results show that both recall and precision are improved by employing the support frequencies.

5. CONCLUSIONS

This paper presents a high precision algorithm for automatic extracting high-frequency words based on statistics, which is aimed to modify the co-occurrence frequencies of patterns in order to eliminate the "bridge-connection" pattern of Chinese characters. The experimental results show that the proposed algorithm can filter out and identify the patterns according to the modified frequencies as well as improve the recall and precision of extracted lexical set. This algorithm can mainly be applied to text categorization and automatic summarization in Chinese.

REFERENCES

[1] ZHANG Chunxia, HAO Tianyong. The State of the Art and Difficulties in Automatic Chinese Word Segmentation (in Chinese). Journal of System Simulation, 2005, 17(1): 138 - 143

[2] HAN Kesong, WANG Yongcheng, CHEN Guilin. Research on Fast High-frequency Strings Extracting and Statistics Algorithm with no Thesaurus (in Chinese). Journal of Chinese Information Processing, 2001, 15(2): 23 - 30

[3] JIN Xiangyu, SUN Zhengxing, ZHANG Fuyan. A Domain-independent Dictionary-free Lexical Acquisition Model For Chinese Document (in Chinese). Journal of Chinese Information Processing, 2001, 15(6): 33 - 39

[4] WU Yingliang, WEI Gang, LI Haizhou. A Word Segmentation Algorithm for Chinese Language Based on N-Gram Models and Machine Learning (in Chinese). Journal of Electronics and Information Technology, 2001, 23(11): 1148 - 1153

[5] GUO Xianghao, ZHONG Yixin, YANG Li. A Fast Algorithm for Chinese Words Automatic Segment Based on Two-letters-word-family Structure (in Chinese). Journal of The China Society For Scientific and Technical Information, 1998, 17(5): 352 - 357

[6] HE Hao, YANG Haitang. Approach of Chinese Document Automatic Classification Based on the Frequency of N-Gram (in Chinese). Journal of The China Society For Scientific and Technical Information, 2002, 21(4): 421- 427