A Comparative Study of the Effect of Word Segmentation On Chinese Terminology Extraction

Luning Ji, Qin Lu, Wenjie Li, YiRong Chen

The Department of Computing, The Hong Kong Polytechnic University,
Hong Kong, China
Email: {cslji, csluqin, cswjli, csyrchen}@comp.polyu.edu.hk

Abstract: Automatic term extraction is the first step towards automatic or semi-automatic update of existing domain knowledge base. Most of the researches applied word segmentation as a preprocessing step to Chinese term extraction. However, segmentation ambiguity is unavoidable, especially in identifying unknown words for Chinese. In this paper, we discuss the effect and limitations of segmentation to Chinese terminology extraction. Detailed study shows that propagated errors caused by word segmentation have great impact on the result of terminology extraction. Based on our analysis and experiments, it is proven that character-based terminology extraction yields much better result than that using segmentation as a preprocessing step.

Keywords: Word Segmentation, Terminology Extraction, Chinese Corpus

1 Introduction

Science and technology has recently generated many new theories, materials, technologies, and concepts within both traditional and novel domains of knowledge. The ever-speedier creation of this knowledge is facilitated and abetted by the wide use of computers and the Internet which allows not only faster speedier and more convenient access to recent knowledge claims but also the continual expansion of the collections of scientific literatures held on the Internet and elsewhere. This domain specific knowledge need to be updated constantly and obviously manual updating which relies on domain experts simply cannot cope with such rapid changes. One potentially useful response to this problem is automatic terminology extraction (TE).

Automatic *term extraction* is the first step in the automatic or semi-automatic extraction of terminology. That is, we must identify terms first before validating that they are domain specific terms, which is the second step called *terminology extraction*. Most research in the area of term extraction has been carried out in European languages, such as English, French and German. Given the well-recognized difficulties associated with segmenting Chinese texts that arise from the non-use delimiters to indicate word boundaries, there are two alternative methods for identifying and extracting Chinese terms: one is to use word segmentation as the preprocessing step, referred to as the *word-based preprocessing model* and the other is to simply identify terms as a string of character patterns no matter what the ultimate number of characters in the term, referred to as the *character-based preprocessing model.* The choice of which method to use is a matter of some importance as incorrect term identification will inevitably result in incorrect terminology identification.

Most of the previous works on Chinese considered word segmentation as the prerequisite step to natural language applications for Chinese. Although previous works in word segmentation have made great progresses, segmentation ambiguity is unavoidable. A key hindrance to segmentation is the the particular difficulty in identifying unknown words for Chinese, which is also directly linked to new term discovery..

This motivated us to look into some language-specific issues: when we build up the model for Chinese terminology extraction, should we use word segmentation? Is the terminology extraction model character-based or word-based? How does word segmentation affect the performance of terminology extraction? How should word segmentation and terminology extraction interact with each other?

In this paper, we discuss these language-specific issues in Chinese terminology extraction. We describe in detail the limitations of word segmentation for use in terminology extraction. We compare character-based and word-based models and show that the precision of segmentation greatly affects the results of terminology extraction and provide some corresponding suggestions on how to solve segmentation problems. We further discuss the interaction of word segmentation and terminology extraction.

The rest of this paper is organized as follows. Section 2 describes the effect of segmentation errors on terminology extraction through the comparisons of a character-based model and a word-based model. Section 3 presents the limitations of segmentation in detail and some feasible suggestions for dealing with these limitations. Section 4 offers our conclusion.

2 Term Extraction Models for Terminology Extraction

Chinese corpus does not have blank space to indicate the word boundaries. The first key issue in Chinese terminology extraction is whether to use the character-based preprocessing model or word-based preprocessing model to extract correct Chinese terms. In this section, we present an automatic term extraction model as an example to extract term candidates which can take input either from the word-base preprocessing model or the character-based preprocessing model. These extracted new term candidates are identified as valid terminology in the second step using terminology extraction algorithms which is not the main focus of this paper. The precision of automatic term extraction directly affects the overall performance of terminology extraction. This is also the reason why in this paper we only show the experimental results on automatic term extraction to analyze the effect of word segmentation.

In this section, we set up our comparative experiments to show the effect of word segmentation on Terminology Extraction (TE). Section 2.1 describes the design of the automatic term extraction model that we will use in our comparative experiments. Section 2.2 describes the experiments setup and results of comparisons between using the two preprocessing models—the character-based model and the word-based model.

2.1 Automatic Term Extraction Model

The term extraction model we use has three parts: filtering of garbage strings, internal measure of candidate terms, and external measure of candidates to determine which of the candidates qualify as valid terms.

Firstly, we apply the PatTree data structure to extract all possible character strings with their frequency counts in the filtering step [2]. Candidate word string list is built and ranked by their frequency and separated by the string length.

Although we do not restrict the patterns of terms, some common non-term patterns which are considered as garbage strings are detected and filtered out. The garbage patterns are identified by a stop-word list, such as "我的(mine)","在香港(in Hong Kong)", "当上课(when having class)", "桌子上(on the table)", "完成了(finished)", because it is unlikely to be an independent term with these stop words at certain specific positions, such as the beginning or the ending.

Secondly, we used two kinds of statistic-based measures to estimate the soundness of an extracted string being a word/phrase [12]: the internal measure and the external measure. Internal measure estimates the soundness by the internal associative strength between constituents of the item [12]. We use the *significance estimation function (SEF)* shown in the following **Equation 1** to measure the internal association. *SEF* is used to judge if a pattern *c* is more complete in semantics than its substrings *a* and *b* where $a \subset c$ and $b \subset c$. *SEF* works especially well for multi-character terms, compared to other commonly-used approaches such as Mutual Information..

$$SEF = \frac{f(c)}{f(a) + f(b) - f(c)} \qquad \textbf{(1)}$$

where *c* ($c = C_1C_2C_3...C_n$) is a lexical pattern to be estimated, *a* ($a= C_2C_3C_4...C_n$)is the longest right substring, and *b* ($b= C_1C_2C_3...C_{n-1}$) is the longest left substring. *f(a)*, *f(b)* and *f(c)* refer to the frequency of occurrence of string *a, b* and *c* in the corpus, respectively.

The value of *SEF* is between 0 and 1.A larger *SEF* indicates that patterns *a* and *b* tend to occur together in the text. Thus *c* is more complete in semantics than either *a* or *b*. *SEF* equals to 1 means that *a* and *b* only occur as substring of *c*. That is to say, the larger *SEF* value is, the more likely the string *c* is a term.

SEF is not very effective for 2 character words bigrams (e.g. "计算calculate") because of potential noise in the counting for single characters as for string *a* ("计") and *b* ("算") where both these characters have very strong ability for form different words such as, character "计" can be contained in words "设计(design)", "统计(statistics)", "计时(time)" and etc. Therefore, for two character bigrams, we propose to use the following formula:

$$SEF\ (c_{bi}) = 1 - \frac{2^{\theta}}{\frac{p(ab)}{p(a)p(b)}} = 1 - \frac{2^{\theta}\, p(a)p(b)}{p(ab)} \qquad \textbf{(2)}$$

where *p(a) p(b)* represents the probability of the substring *a* and *b* respectively which forms the 2 character word *ab*. *θ* is an experimentally decided parameter. The maximum value of $SEF(c_{bi})$ is 1, which means the pattern *ab* has high possibility to be a word.

As a larger value of *SEF* implies a stronger associative strength in a string, it is more likely that the string is a term. Thus, the criterion of judgment is very simple: The candidate string would be accepted as a term if its internal associative strength is larger than a given threshold.

The external measure, which estimates the soundness by the dependency of the item on its context as its external strength, makes use of the contextual information of the candidate string [12]. We apply the *C-value* measure to calculate the external strength [1][7]. The *C-value* is given as follows:

$$C\_value\ (c) = \begin{cases} \log_2 |c|\, f(c) & \text{c is not nested} \\ \log_2 |c|\,(f(c) - \frac{1}{P(T_C)} \sum_{b \in T_c} f(b)) & \text{c is nested} \end{cases} \qquad \textbf{(3)}$$

where *c* is the candidate string, |*c*| is the number of words in string *c; f*(*c*) is the frequency of occurrence of *c* in the corpus; T_c is the set of extracted candidate terms that contain *c*; $P(T_c)$ is the number of these longer candidate terms. *C-value* aims to further measure the candidacy of a term especially those nested terms on their independency. The formula shows that the more often a candidate string occurs alone and the longer its size, the higher the *C*-value. The more often a candidate string occurs as a substring, the lower its value. Furthermore, the more the number of longer strings in which the candidate string occurs, the higher its value [4]. Simply put, a larger value implies a more likely term. For example, the word "计算"(calculate) can be used as an independent term, yet it can also be contained in another term "计算机"(computer). Let us take word "贝叶斯"(Bayes) as another example. Although the internal measure of "贝叶斯" is very high, its C-value is low, since "贝叶斯" always occurs as substring of the words "贝叶斯定理"(Bayes' Theorem), "贝叶斯算法"(Bayes Algorithm) or "贝叶斯决策"(Bayes Decision). That is to say, the word "贝叶斯" (Bayes) is less likely to be an independent term. In our term extraction model, only candidates with both the internal and external measures larger than certain thresholds are considered as valid terms.

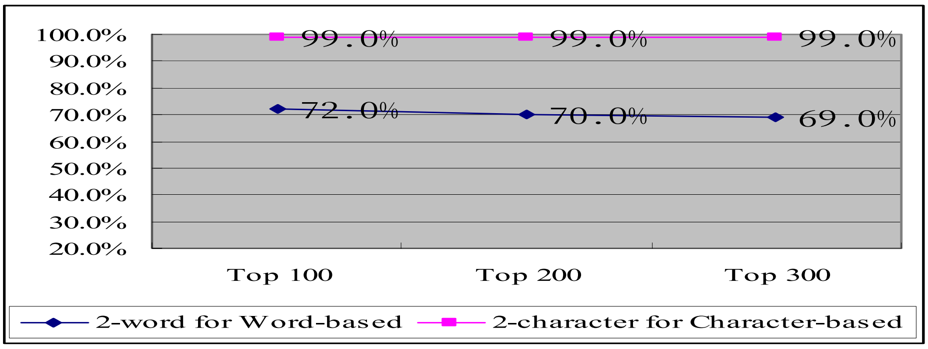
2.2 Character-based Preprocessing Model & Word-based Preprocessing Model

To evaluate the effect of segmentation, we use a character-based preprocessing model and a word-based preprocessing model for the term extraction system. The difference between the character-based model and the word-based model is the definition of unit: a unit in the character-based model refers to one character, compared to one word in word-based model. As the target is domain specific extraction, we select 16 papers from 16 *IT* journals as our testing corpus, which has a total of 1,500,000 characters. Both models are tested on this same data set. For the word-based model, we applied the

Unicode-based adaptive segmenter [10] first. This segmenter relies on dictionaries and includes five algorithms: simple dictionary matching, forward minimum matching, forward maximum matching, backward minimum matching, and backward maximum matching. The segmentation system was tested on a six month sample of data from a newspaper, the People's Daily, and got the 94.1% precise with a recall rate of 94.8%, and *F 1* of 94.4%. PoS is 96.0% precise, with a recall rate of 94.9%, and an *F 1* of 95.4% [10]. At training time, the word boundaries are consistent with known word boundaries. At test time, however, the segmenter can create words which do not agree with the gold-standard word boundaries which refer to the existing known word boundaries [3].

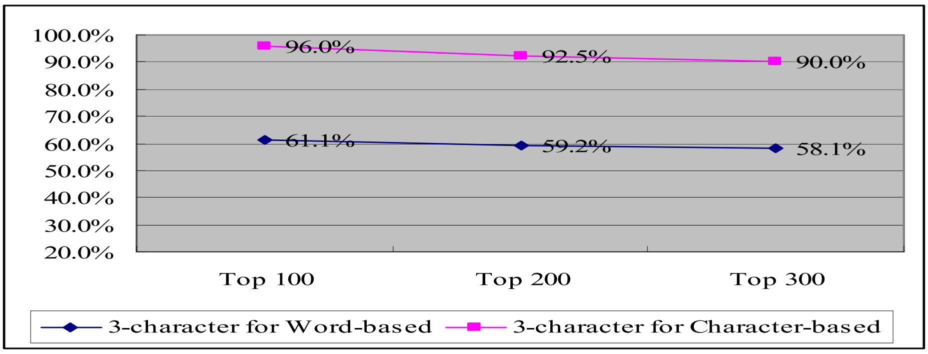
Due to the limitation of this paper, the experiments on parameter selection itself are not discussed. We take the parameter values of C_value > 5, SEF> 0.5 and θ in formula $SEF(C_{bi})$ as 5.5. For the Character-based model, a total of 1,459 bi-grams, 1,184 tri-grams, and 5,438 quadra-grams, in which the gram refers to the number of characters, are extracted. For the word-based model, in which the gram refers to the number of words, a total of 7401 uni-grams (at least two characters), 17,706 bi-grams, and 12,808 tri-grams using the same set of parameters are extracted.

Figure 1 and **Figure 2** show the comparisons of the two models. The comparisons are presented in two ways—the same unit-length (unit in terms of grams thus most likely different character lengths) and the same character string length.



Top Words	2-word for Word-based	2-character for Character-based
Top 100	72.0%	99.0%
Top 200	70.0%	99.0%
Top 300	69.0%	99.0%

Fig. 1. The precision of top 100, 200, 300 words for 2-Word of word-based and character-based model



Top Words	3-character for Word-based	3-character for Character-based
Top 100	61.1%	96.0%
Top 200	59.2%	92.5%
Top 300	58.1%	90.0%

Fig. 2. The precision of top 100, top 200, top 300 words of 3-Length term for both word-based and character-based model

From the above two figures, we can see that the precision of character-based model is much higher than the word-based model using the same comparison standards. Because we apply the same internal and external measures, the lower precision must be caused by the performance of segmentation. We find that some terms are fragmented because of wrong segmentation. Besides, these wrongly-segmented words are wrongly combined with its neighboring terms so that the segmentation error is propagated. For example, the phrase "时间差和声级差"(Time difference and sound grading) is wrongly segmented as "时间差/n 和声/n 级差/n "(time difference/n harmonic/n grading/n). The term "声级差" (sound grading) is segmented and the character "声(sound)" is wrongly combined with the character "和 (and)"to form the wrong word "和声"(harmonic). It should be pointed out that no matter how good a segmentation algorithm is, it is normally used as a general purpose segmentor which is usually not tailored for a specific domain. This means that the ability of segmentors to deal with unknown or new words, which is needed in term extraction, especially new term extraction, is very limited. We will discuss these limitations in detail in the next section.

Even though the above experiments show that the word-based model performs worse than that of the character-based model, a more careful examination reveals that the word-based model performs better for certain word patterns. For example, the word-based model performs better on the organization category which is tagged as "/nt" and the place category which is tagged as "/ns". Organization names always have many characters, for example "北京大学(Peking University)" contains four Chinese characters. As the segmentor can analyze the larger context to determine the word boundaries and its tagging, the word-based model has larger probabilities to make a correct decision, for example, to mark "北京大学/nt"(Peking University/nt) as "/nt". Even if some very long organization names cannot be recognized correctly as a complete word, the combination pattern of a place or an organization always leads to a new complete organization name. For example, if the segmentation result is "香港/ns 理工大学 /nt "("Hong Kong/ns Polytechnic University/nt"), we can still identify "香港理工大学"(The Hong Kong Polytechnic University) as one word. In addition, the word-based model performs better on English words contained in the Chinese corpus. The *IT* domain corpus always has large amount of English terms, as well as some combined words of Chinese and English. So the ability to recognize correct English words is also important. The word-based model can recognize each independent English word and segment them correctly while the character-based model can only recognize each character one by one. It is hard to recombine the segmented characters into words. However, if the character-based algorithm can first identify English words as single unit, the results of both models for English should be the same.

3 Effect of Word Segmentation

Our final target is to extract domain-specific terminology to update the existing domain knowledge base. The above experiments show that the performance of segmentation really affects the final performance of terminology extraction as it affects the precision of forming the candidate term list, which is the first step of terminology extraction. Aiming at terminology extraction, three main limitations of word segmentation are discussed in detail in this section.

3.1 Segmentation Ambiguity

Although resolving segmentation ambiguities for Chinese word segmentation has received considerable attention in the previous years, none of the segmentors can promise to handle all the ambiguities. Because of the complexity of Chinese syntax and semantics, word segmentation ambiguities cannot be avoided, which is also the main reason that causes the inaccuracy of segmentation. Word segmentation ambiguities can be roughly classified into two classes: overlapping ambiguity (OA), and combination ambiguity (CA) [8].

Liang (1987) defines OA and CA in detail as follows:

Definition 1 *A character string ABC is called an* ***overlap ambiguity string*** *(OAS) if it can be segmented into two words either as AB/C or A/BC (not both) depending on different context.*

Definition 2 *A character string AB is called a* ***combination ambiguity string*** *(CAS), if A, B, and AB are words.*

Table 1. Segmentation Error Types of Top 500 Extracted Words

Segmentation Error Type	OAS	CAS	Personal Name	Other	Total
Number	62(39.5%)	48(30.6%)	17(10.8%)	30 (19.1%)	157
Percentage in Total Extracted	12.4%	9.6%	3.4%	6.0%	31.4%

Table 1 shows the segmentation error types of the Top 500 extracted words of word-based model in the experiments mentioned in Section 2. Among the total of 157 wrong strings, OAS and CAS nearly takes a combined 70%. Since the segmentation ambiguity is inevitable, what we should do for terminology extraction is try to find the missed terms to mitigate the propagated error caused by segmentation. As introduced in Section 2, our automatic term extraction method tries to use both internal and external measures to resolve the wrongly-segmented words especially the errors caused by segmentation ambiguities. The external measure, which estimates the soundness by the dependency of the item on its context, helps to decide whether the term is an independent word or a substring in a longer term. This measure helps to solve some combination ambiguity errors. For example, if the candidate term "串行/b 程序/n"(parallel process), is evaluated by the external measure, we can combine them back to get the correct term "串行程序"(parallel process). However, not all the combination ambiguity errors can be detected. Sometimes, we may wrongly combine the words in segmentation. For example, in the segmented sentence "学习/v 中长期/d 积累/n"(long time accumulation through the study), we can see that the words "中"(in) and "长期"(long term) are wrongly combined together.

It is more difficult to resolve the overlapping ambiguity errors even if we apply the internal measure to estimate whether the candidate term is complete in semantics. According to our experiments statistics, in the top 500 extracted words in the word-based model, there are 62 wrong words because of overlapping ambiguity error among a total of 157 wrong words, taking nearly 40%. In other words, the overlapping ambiguity error causes more than 10% reduction on the precision. For example, the sentence"不可避免(unavoidable)/l 地带/n(zone) 来(come)/v" wrongly selects "地带" as a independent word. Take the sentence "而已有方法则是"("and the existing method is") as another example. The segmentation result is "而已(nothing more)/y 有方(in the right way)/v 法则/n(rule) 是(is)/v." There are two consecutive overlapping ambiguity errors in this sentence—"而已有" and "方法则". Since "方法(way)" has been divided into two characters and each character is combined with other characters to form other words, it is hard to recover the word"方法(way)". Most of the serious errors by overlapping ambiguity happen this way—the wrongly-segmented words are combined with it neighboring words, and bring on another overlapping ambiguity error for its neighboring words. This kind of error chain cannot be detected easily.

Another consequence by segmentation ambiguity is the PoS tagging error. Some researches extract certain language patterns as the candidate terms according to the PoS tags for terminology extraction. There are also approaches which calculate the termhood of candidate terms in terminology extraction by taking syntactic information, which is also represented by PoS tags. Therefore, the wrong PoS tags will also lead to propagated on the final performance of terminology extraction.

3.2 Granularity of Word Segmentation

Upon examining the testing data, we discover that the granularity of segmentation also affect the performance of the word-based model. The segmentor used in our experiments has options for us to select on granularity of words—High level, which means the longest word is selected when deciding the word boundaries and Low level, which means the shortest word is selected. Each option performs well only for certain patterns. For example, the High granularity performs better on personal names and organization names. A personal name including a family name and a first name can be recognized as a complete word when longest word is selected. So that we can easily filter out these personal names which are tagged as "/nr" since personal names are not the terms that we are interested on terminology extraction. Organization names are not the target for terminology extraction either. As we mentioned above, many organization names contain many characters. The high granularity helps to solve the combination ambiguity on organization category. However, high granularity may introduce some segmentation errors especially for unknown words and new terms. It may wrongly combine two independent words together, which is hard to be separated then. On the contrary, if we apply the low granularity, the unknown words may be divided dispersedly. By evaluating the external measures, we can combine them back. For instance, the new term "鲁/Ng 棒/a 性/Ng" has been segmented as three characters. After we found the strong association between these three characters, we can combine them back to form the correct new term "鲁棒性"(robustness).

According to the above analysis, for our final target terminology extraction, we prefer to use high granularity to detect personal names as well as organization names, and use low granularity to other word patterns, especially the unknown words and new words.

3.3 Domain Specificity

As we have mentioned earlier, no matter how good segmentation is, a segmentor is normally used as a general purpose segmentor which is seldom tailored for a specific domain. This means that its ability to deal with unknown words or new words, which is needed in term extraction, especially new term extraction, is very limited. In fact, unknown/new words are also the main reason that brings the segmentation ambiguity. A better way to solve this domain specificity problem is to use a domain-specific segmentor for word segmentation which requires, the segmentor be trained by domain specific segmented corpus and domain-specific dictionaries or lexicons for matching. However, domain specific corpora are relatively small and it may not be sufficient to train the segmentor to work well. The training can also be time consuming. The shortage of domain-specific resources and the need for timely update are in fact the main motivations to develop terminology extraction techniques. Actually, term extraction result can help to improve the performance of segmentation. Thus, they are inter-dependent.

One suggestive solution to solve this domain specificity problem is to make use of the preliminary result of terminology extraction without segmentation, and then add the newly identified terminologies into the general dictionaries. In this way, the general dictionary used for word segmentation will contain more domain specific terms. So its ability to handle unknown words or new words in these domains can be improved. We can repeat the process of terminology extraction with the updated segmentor to extract more precise terminology in selected domain. Consequently, the improvement of segmentation performance can lead to a better performance on terminology extraction, which in turn helps to further improve the performance of segmentation.

4 Conclusion

Even though word segmentation seems to be the common practice as the preprocessing step for natural language processing applications for Chinese, they can also cause problems as mistakes made in segmentation can be propagated to affect the performance in the subsequent applications. Segmentation errors in particular can affect the performance of terminology extraction greatly because of its limited ability of segmentors to deal with unknown words which may happen to be the new words we are looking for in terminology extraction.

In this paper, we have discussed the effect of segmentation errors on terminology extraction. Comparisons between character-based and word-based preprocessing models prove that the precision of segmentation can affects the results of terminology extraction greatly. Three main limitations of segmentation on terminology extraction are described in detail including segmentation ambiguity, the granularity of segmentation, and domain specificity. Some corresponding suggestions on how to solve the segmentation problems are also provided. In conclusion, one feasible way is to make use of the preliminary result of terminology extraction without segmentation, and add these newly-found terminologies into the general dictionaries for the segmentor. The updated segmentor can perform better on domain specific words, which can produce better result on terminology extraction. This kind of mutual benefit can really be useful for Chinese natural language processing. Consequently, it is a good way to integrate word segmentation and terminology extraction. We have started our research work based on this idea.

It is not our intention to claim that segmentation can be removed for Chinese language processing applications. Yet it is proven that for terminology extraction in a specific domain, word segmentation can be avoided as term exaction algorithms can determine whether a string pattern is more likely a term and further confirmed by terminology extraction algorithm to identify whether it is indeed domain specific (and thus terminology).

Acknowledgments. The project is supported by the Hong Kong Polytechnic University funded project B-Q824 with account number RGBN and account number RGED.

References

1. E. Milios, Y. Zhang, B. He, and L. Dong. "Automatic Term Extraction and Document Similarity in Special Text Corpora". In *Proceedings of the Sixth Conference of the Pacific Association for Computational Linguistics* (PACLing'03), 2003, pages 275-284, Halifax, NS, Canada, August 22-25, 2003.
2. Chien, L.F. "Pat-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval". *Information Processing and Management,1999,* vol.35 pp.501-521
3. H Jing, R Florian, X Luo, T Zhang, A Ittycheriah, "HowtogetaChineseName(Entity): Segmentation and Combination Issues", *Proceedings of EMNLP'03, 2003*
4. Ismail Fahmi, "C-value method for multi-word term extraction", In seminar in Statistics and Methodology, May 23, 2005
5. Kageura, K. and Umino, B. "Methods of automatic term recognition: a review". *Terminology* 3(2), 259-289, 1996.
6. Katerina T. Frantzi, "Incorporating Context Information for the Extraction of Terms". In Proc. of *ACL/EACL '97*, pages 501-503, Madrid, Spain, July 1997
7. K.T.Frantzi and S. Annaniadou, "Extracting nested collocations", In the Proc. Of COLING '96, pp.41-46 (1996)
8. Liang, Nanyuan. (梁南元). 书面汉语自动分词系统— CDWS (A written Chinese automatic segmentation system – CDWS). In: 中文信息学报 (*Journal of Chinese Information Processing*). 1987, 2: 44-52.
9. Munpyo Hong, Sisay Fissaha, Johann Haller, "Hybrid Filtering for Extraction of Term Candidates from German Technical Texts", *Terminology and Artificial Intelligence TIA'2001*
10. Qin Lu, Shiu-tong Chan, Baoli Li and Shiwen Yu, "A Unicode-based Adaptive Segmenter", *Journal of Chinese Language and Computing 14 (3):221-234,2004*
11. Schone, P., Jurafsky D. "Is knowledge-free induction of multiword unit dictionary headwords a solved problem?": *In proceedings of EMNLP 2001.*
12. Shengfen Luo, Maosong Sun. "Two-Character Chinese Word Extraction Based on Hybrid of Internal and Contextual Measures": *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, July 2003, pp. 24-30.*
13. Zhifang SUI Yirong CHEN Zhouchao WEI, "Automatic recognition of Chinese scientific and technological terms using integrated linguistic knowledge", Proc. of the *Int. Conf. on Natural Language Processing and Knowledge Engineering*, 2003.
14. Zhang Hua-Ping, Qun Liu, Hao Zhang and Xue-Qi Cheng. "Automatic Recognition of Chinese Unknown Words Based on Role Tagging", *1st SIGHAN Workshop* 2002: 71-77