Chinese Term Extraction Based on Delimiters

Yuhang Yang[12], Qin Lu[2], Tiejun Zhao[1]
[1] School of Computer Science and Technology, Harbin Institute of Technology
[2]Department of Computing, Hong Kong Polytechnic University
E-mail: 1983yang@gmail.com, csluqin@comp.polyu.edu.hk, tjzhao@mtlab.hit.edu.cn

Abstract

Existing techniques extract term candidates by looking for internal and contextual information associated with domain specific terms. The algorithms always face the dilemma that fewer features are not enough to distinguish terms from non-terms whereas more features lead to more conflicts among selected features. This paper presents a novel approach for term extraction based on delimiters which are much more stable and domain independent. The proposed approach is not as sensitive to term frequency as that of previous works. This approach has no strict limit or hard rules and thus they can deal with all kinds of terms. It also requires no prior domain knowledge and no additional training to adapt to new domains. Consequently, the proposed approach can be applied to different domains easily and it is especially useful for resource-limited domains. Evaluations conducted on two different domains for Chinese term extraction show significant improvements over existing techniques which verifies its efficiency and domain independent nature. Experiments on new term extraction indicate that the proposed approach can also serve as an effective tool for domain lexicon expansion.

1. Introduction

Terms are the lexical units to represent the most fundamental knowledge of a domain. Term extraction involves two steps. The first step extracts candidates by unithood calculation and the second step verifies them as terms measured by termhood (Kageura and Umino, 1996). *Unithood* measures the strength which qualifies a string as a valid term. *Termhood* measures the degree at which a term represents some domain specific concept. This study focuses on unithood measures for term candidate extraction only.

Existing techniques extract term candidates using two kinds of statistic based measures including internal association (e.g. Schone and Jurafsky, 2001) and context dependency (e.g. Sornlertlamvanich et al., 2000). These techniques are also used in Chinese term candidate extraction (e.g. Chen et al., 2006; Ji and Lu, 2007). All the current techniques focus on domain dependent terms and use a weighted approach to consider various features to identify term boundaries. However, only one or two features are useful in a particular instance. The algorithms always face the dilemma that fewer features are not enough to distinguish terms from non-terms whereas more features lead to more conflicts among selected features. In practice, they all suffer from two major problems. The first problem is that the algorithms cannot identify certain kinds of terms. These statistics based techniques are very sensitive to term frequency and terms with low frequencies cannot be extracted. In order to achieve a reasonably good precision, most techniques have strict limits on the maximal length of the extracted terms which can compromise the identification of long compound terms. With the use of predefined rules to weed out noises, some techniques also weed out useful terms. The second major problem is that most techniques must use full segmentation for Chinese term extraction which is usually less successful to handle domain specific data. Chinese segmentation algorithms do have good performance on general purpose data. Yet, they need to be trained to work in a specific domain and the identification of the terms can be considered as part of the training process. Thus, relying on segmentation to train the segmentation is obvious not going to work well (Huang et al., 2007).

In this paper, term candidate extraction is considered in a totally different way and a novel approach is proposed to overcome existing problems. Instead of looking for features associated with domain specific terms, term candidates are extracted by identifying the relative stable and domain independent boundary marker kind of words immediate before and after these terms. In contrast to previous researches, the proposed approach does not have strict limits on frequency or length and thus it can identify low frequency and compound terms. Secondly, it requires no full segmentation and thus there are no cascading segmentation errors in term extraction. Also, the proposed approach extracts term candidate by identifying boundary markers which are quite domain independent. It requires no prior domain knowledge and no adaptation for another domain. Thus, they can be applied to different domains easily. It is especially useful in resource-limited domains.

The evaluation of this work is based on the experiments conducted for Chinese in two different domains, the IT (information technology) domain and the legal domain. Results show that term extraction using the proposed method achieves quite significant improvements over previous algorithms. Two sets of experiments also verify its domain independent nature which indicates that the technique developed can be applied to other domains. Another set of experiments on new term extraction shows that the proposed approach can serve as a much better tool to identify new terms in a domain and thus can serve as an effective tool in domain lexicon expansion.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 describes the methodology and the algorithms. Section 4 presents the experiments and evaluations. Section 5 is the conclusion.

2. Related work

In general, there are two kinds of statistic-based measures (Luo and Sun, 2003) for estimating the unithood of a term candidates. The first kind is the internal measure which estimates the strength by the internal associative measures between constituents of the candidate characters. Some limited statistical information on the occurrence probability of the whole unit and its component elements are mainly used in these algorithms. Nine widely adopted internal measures, such as *frequency* and *mutual information*, are listed in (Schone and Jurafsky, 2001). The second kind is the contextual measure which estimates the strength by the dependency of the candidate on its context using measures such as the *left/right entropy* (Sornlertlamvanich et al., 2000), the *left/right context dependency* (Chien, 1999), and *accessor variety criteria* (Feng et al., 2004).

Most previous studies use one or both of them for unithood calculation. The *UnitRate* algorithm proposed in (Chen et al., 2006) integrates occurrence probability and marginal variety probability of the candidates and all its components. The *TCE_SEF&CV* algorithm presented in (Ji and Lu, 2007) applies the significance estimation function and *C*-value measure (Frantzi et al., 2000) to estimate the internal and external strength for unithood calculation. However, these algorithms do not perform well for low frequency terms and long terms because of data sparseness. They also applied full segmentation which normally does not perform well in domain specific corpora and can have cascading errors on the term extraction results.

3. Methodology

Generally speaking, sentences are constituted by substantives and functional words. Domain specific terms (*terms* for short) are more likely to be domain substantives. Words immediate before and after these terms, called *predecessors* and *successors* of the terms, are likely to be either functional words or other general substantives connecting terms. In fact, these predecessors and successors can be considered as markers of terms, and are thus referred to as *term delimiters* (or simply *delimiters*) in this paper.

In contrast to terms, delimiters are mainly functional words and general substantives which are relatively stable and domain independent. Thus they can be extracted more easily. Instead of looking for features associated with domain specific terms in other works, this paper looks for features associated with term delimiters. In a way, terms are identified by finding their predecessors and successors as term boundary markers. Words between term boundaries are then considered as term candidates.

The following gives two example sentences:

(1) 扫描隧道显微镜是一种基于量子隧道效应的高分辨率显微镜 (Scan tunneling microscope is a kind of quantum-tunneling-effect based high angular resolution microscope)

(2) 社会主义制度是中华人民共和国的根本制度 (Socialist system is the basic system of the People's Republic of China)

In sentence (1), “扫描隧道显微镜”(scan tunneling microscope), “量子隧道效应”(quantum-tunneling-effect) and “高分辨率显微镜”(high angular resolution microscope) are IT domain terms whose boundaries can be determined by the delimiters “是”(is), “一种”(a kind of), “基于”(based) and “的”(adjective marker). In sentence (2), “社会主义制度”(the socialist system), “中华人民共和国”(People's Republic of China) and “根本制度”(basic system) are legal domain terms whose boundaries can be determined by the delimiters “是”(is) and “的”(adjective marker). Even though sentence (1) and sentence (2) are from different domains, the words such as “是”(is) and “的”(adjective marker) occur as delimiters in both of them, which indicates that they are domain independent. The delimiters occur immediately before or after terms in both sentences. In other words, their usage and locations are stable and thus can be identified as term boundary markers.

The proposed delimiter identification based algorithm, referred to as *TCE_DI* (Term Candidate Extraction – Delimiter Identification), extracts term candidates from a domain corpus, $Corpus_{extract}$, by using a delimiter list, referred to as the *DList*. Given a *DList*, the algorithm TCE_DI_{DList} itself is quite straight forward. For a given character string *CS* ($CS = C_1C_2\ldots C_n$) in $Corpus_{extract}$, as shown in Figure 1, where each C_i is a Chinese character. Suppose there are two delimiters $D_1 = C_{i1}\ldots C_{il}$ and $D_2 = C_{j1}\ldots C_{jm}$ in *CS* where $D_1 \in DList$ and $D_2 \in DList$. The string *CS* is then segmented to five substrings: $C_1\ldots C_{ib}$, $C_{i1}\ldots C_{il}$, $C_{ia}\ldots C_{jb}$, $C_{j1}\ldots C_{jm}$, and $C_{ja}\ldots C_n$. Since $C_{i1}\ldots C_{il}$ and $C_{j1}\ldots C_{jm}$ are delimiters, $C_1\ldots C_{ib}$, $C_{ia}\ldots C_{jb}$, and $C_{ja}\ldots C_n$ are regarded as three term candidates (TC_1, TC_2 and TC_3 in Figure 1). If there is no delimiter contained in *CS*, the whole string $C_1C_2\ldots C_n$ is regarded as a term candidate.

Figure 1: Paradigm of term candidate extraction

The *DList* can be obtained either from a delimiter training corpus or from a given stop word list. Given a delimiter training corpus, referred to as $Corpus_{D\_training}$, normally a domain specific corpus, and a domain lexicon $Lexicon_{Domain}$, the *DList* can be obtained based on the

following algorithm, referred to as *DList_Ext*.

Step 1: For each term T_i in $Lexicon_{Domain}$, mark T_i in $Corpus_{D\_training}$ as a non-divisible lexical unit. For example, in the sentence "微电子技术引发了本世纪的信息革命"(Microelectronic technique has triggered the information revolution of this century), the two IT domain terms "微电子技术"(Microelectronic technique) and "信息革命"(information revolution) are marked as non divisible lexical units because they are terms in $Lexicon_{Domain}$.

Step 2: Segment the remaining text in the corpus. In this instance, "引发了本世纪的" are segmented into "引发(triggered) 了(past tense marker) 本世纪(of this century) 的(adjective marker)".

Step 3: Extracts predecessors and successors of all T_i as delimiter candidates. The predecessor "的" of "信息革命" and the successor "引发" of "微电子技术" are extracted as delimiter candidates.

Step 4: Remove delimiter candidates that are contained in an existing term T_i.

Step 5: Rank the candidates by frequency and the top N_{DI} number of items are considered delimiters. N_{DI} is an algorithm parameter to be determined experimentally.

The *DList_Ext* algorithm basically use the known terms given by $Lexicon_{Domain}$ to find the delimiters. It can be shown in the experiments later that $Lexicon_{Domain}$ does not need to be comprehensive. If even a small training set, $Corpus_{D\_training}$, is not available in a language without sufficient domain specific NLP resources, a stop-word list produced by experts or from a general corpus can also serve as the *DList* without using the *DList_Ext* algorithm.

4. Experiment and Discussion

4.1 Data Preparation and Performance Measurements

To conduct the experiments for Chinese, four separate corpora of different domains in different sizes are used. The first set, referred to as $Corpus_{IT\_small}$, contains 16 papers of 77K in size from Chinese IT journals between 1998 and 2000. $Corpus_{IT\_Small}$ is used as training data to obtain the delimiter list of IT domain, $DList_{IT}$, according to the *DList_Ext* algorithm given in Section 3. The second set, referred to as $Corpus_{IT\_Large}$, contains 433 papers of 6.64M in size from the Chinese IT journal "Journal of Software" between 1998 and 2000. $Corpus_{IT\_Large}$ is used to evaluate the proposed algorithm. In order to validate that the algorithm works for different domains, a third corpus is taken from the legal domain, referred to as $Corpus_{Legal\_Small}$, which contains 9 Chinese criminal law articles of 344K in size for the laws enacted between 1999 and 2006. $Corpus_{Legal\_Small}$ is used as training data to obtain the delimiter list of legal domain, $DList_{Legal}$, extracted according the proposed *DList_Ext* algorithm. The forth set, $Corpus_{Legal\_Large}$, used as test data, contains 83 Chinese law articles of 1.04M in size for the laws enacted between 1982 and 2005. Two domain lexicons, referred to as $Lexicon_{IT}$ and $Lexicon_{Legal}$, are obtained manually from the two training corpora $Corpus_{IT\_small}$, and $Corpus_{Legal\_Small}$, respectively. $Lexicon_{IT}$ contains a total of 3,337 IT terms which are extracted from $Corpus_{IT\_small}$ and verified manually. $Lexicon_{Legal}$ contains a total of 394 legal terms which are extracted from $Corpus_{Legal\_Small}$ and also verified manually.

To verify that the approach works for delimiter lists that are not necessarily generated from domain specific corpora, the evaluation also uses a stop word (*SW*) list, denoted as $DList_{SW}$, which contains 494 general purpose stop words downloaded from a Chinese natural language processing resource website (www.nlp.org.cn) without any modification.

Experiments for term extraction are conducted on two different domains. $Corpus_{IT\_Large}$ is used as test data for IT domain. A lexicon, $Lexicon_{PKU}$, is used as standard term set for evaluation on the IT domain. $Lexicon_{PKU}$ contains a total of 144K manually verified IT terms supplied by the Institute of Computational Linguistics, Peking University. The performance is evaluated in term of precision according to the follow formula:

$$precision_{TE} = \frac{N_{Lexicon} + N_{New}}{N_{TCList}} \quad (1)$$

Where N_{TCList} is the total number of extracted candidates in the term candidate list *TCList*, $N_{Lexicon}$ denotes the number of extracted term candidates in *TCList* which are also in $Lexicon_{PKU}$, N_{New} denotes the number of extracted term candidates that are not in $Lexicon_{PKU}$, yet are considered correct. They are thus considered newly identified terms with respect to $Lexicon_{PKU}$. It should be pointed out that, in principle, the verification of all the new terms should be done manually. However, manual verification of all the experimental data is not possible since the test data set is quite large. So, a sampling technique is used in which one sample is selected for every 10 extracted terms. Thus 500 samples for the top 5,000 extracted terms are used for evaluation. In the second set of experiments on $Corpus_{Legal\_Large}$, the same sampling is used except that there is no standard legal term list available. Thus, $N_{Lexicon}$ is not considered.

To compare the ability of different algorithms in identify new terms, that is, terms outside of the lexicon list, another measurement is applied to $Corpus_{IT\_Large}$ against the domain lexicon based on the following formula:

$$R_{NTE} = \frac{N_{New}}{N_{TCList}} \quad (2)$$

Where *TCList* and N_{New} are the same as given in formula

(1). A higher R_{NTE} indicates more extracted terms are outside the lexicon list and is thus considered new terms. Similar to the measurements of out of vocabulary (OOV) in Chinese segmentation, R_{NTE} shows the ability of the algorithms to identify new terms. The newly identified terms can be used for domain knowledge update including lexicon expansion.

4.2 Evaluation of Delimiter List Extraction

In order to determine the algorithm parameter N_{DI} for *DList_Ext* so that the extracted *DList* can have a good coverage, Table 1 shows experiments on the sentence coverage of the top ranked delimiters $DList_{IT}$, $DList_{Legal}$, and $DList_{SW}$ in different ranges on the test corpora $Corpus_{IT\_Large}$ and $Corpus_{Legal\_Large}$, respectively. The sentence coverage denotes the percentages of sentences containing delimiters. Since $DList_{IT}$ are extracted from $Corpus_{IT\_Small}$, the sentence coverage of $DList_{IT}$ on $Corpus_{IT\_Large}$ is marginally higher than that on $Corpus_{Legal\_Large}$. The sentence coverage of $DList_{legal}$ on $Corpus_{Legal\_Large}$ is also marginally higher than that on $Corpus_{IT\_Large}$. The sentences which do not contain delimiters are mainly short sentences or general sentences which contain less domain information.

	$Corpus_{Legal\_Large}$ (11,048 sentences)	$Corpus_{IT\_Large}$ (60,508 sentences)
$DList_{IT}$ (Top100)	77.6%	89.1%
$DList_{IT}$ (Top300)	84.6%	92.6%
$DList_{IT}$ (Top500)	90.3%	93.4%
$DList_{IT}$ (Top700)	92.7%	93.9%
$DList_{legal}$ (Top100)	95.8%	92.6%
$DList_{legal}$ (Top300)	97.8%	96.2%
$DList_{legal}$ (Top500)	98.7%	96.8%
$DList_{legal}$ (Top700)	99.1%	97.1%
$DList_{SW}$	98.1%	98.1%

Table 1: Coverage of Delimiters on Different Corpora

It is obvious that higher coverage can be achieved when more delimiters are included. However, the significance of the improvement slows down once N_{DI} reaches 500.

To further determine a good cut off point N_{DI}, for the delimiter list, a frequency analysis is also conducted as shown in Figure 2. The frequencies of the top ranked delimiters are much higher than those in the lower ranks. Taking frequencies of $DList_{IT}$ on $Corpus_{IT\_Large}$ as an example, the average frequency of the top 100 delimiters is 1,221.4 which is more than 13 times that of the top 500 to 700. The results coincide with the results shown in Table 1 where improvement becomes insignificant after the top 500. Thus, it is reasonable to take 500 for N_{DI}. In fact, the experiments to be discussed later in Figure 3 to Figure 6 further confirm this. The fact that the distributions of the delimiters in different domains have similar trend also indicates that extracted delimiters are domain independent and stable.

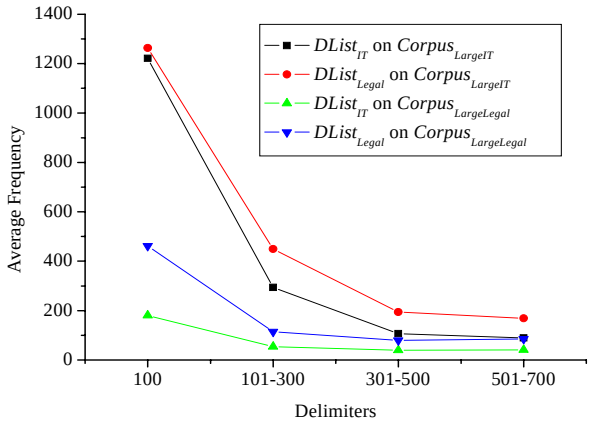

Figure 2: Frequency of Delimiters on Domain Corpora

Two sets of experiments on $Corpus_{IT\_Large}$ and $Corpus_{Legal\_Large}$ are conducted to compare the performance of the proposed algorithm *TCE_DI* by using different ranges of top ranked delimiters as shown in Figure 3 to Figure 6.

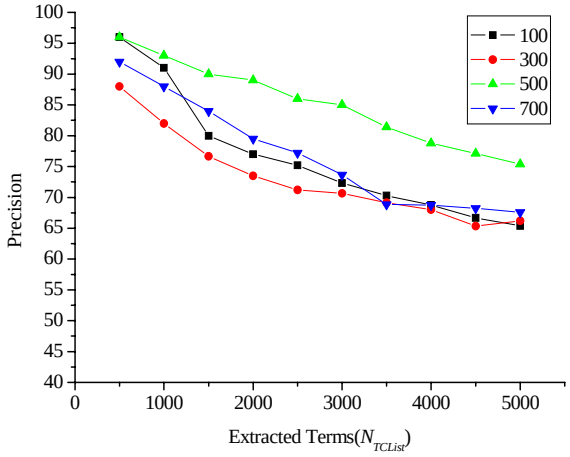

Figure 3: Performance of $DList_{IT}$ on $Corpus_{IT\_Large}$

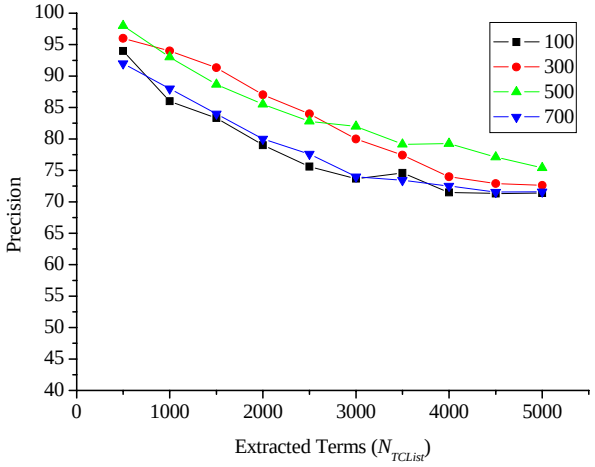

Figure 4: Performance of $DList_{Legal}$ on $Corpus_{IT\_Large}$

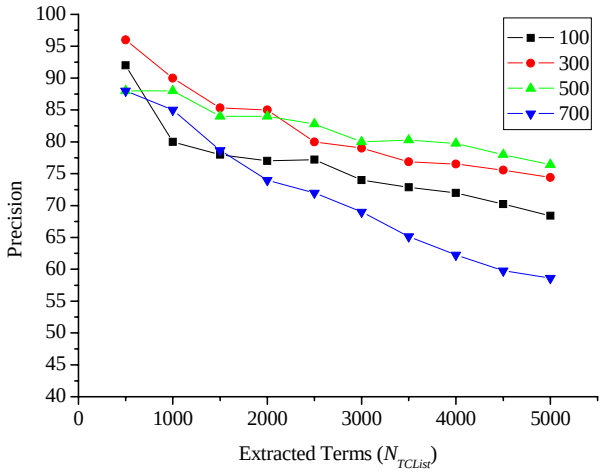

Figure 5: Performance of $DList_{IT}$ on $Corpus_{Legal\_Large}$

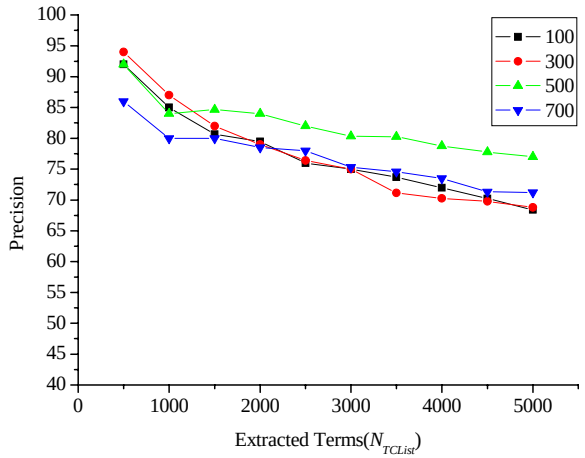

Figure 6: Performance of $DList_{Legal}$ on $Corpus_{Legal\_Large}$

In Figure 3 to Figure 6, N_{DI} = 500 is the best performer. Fewer delimiters are not enough to identify the term boundaries. For example, "改进计算效率"(improve computational efficiency) composed of a general word "改进"(improve) and a IT term "计算效率"(computational efficiency) is considered a IT term by mistake. Because "改进" is not on the $DList_{IT}$ when N_{DI} = 100. When N_{DI} = 500, "改进" is contained in $DList_{IT}$. Thus, the term boundary "计算效率" is identified accurately. On the other hand, too many delimiters may include some noise that would split some terms into pieces. For example, an important IT domain terms "存取"(access) is added to $DList_{IT}$ when N_{DI} = 700. Hence some IT terms which contain "存取" as a component such as "媒体存取层"(media access layer) are split and the retained pieces such as "媒体"(media) and "层"(layer) are considered IT terms by mistake. Based on this set of experiments, N_{DI} = 500 is chosen as the cut off point for both $DList_{IT}$ and $DList_{Legal}$ in the subsequent evaluations.

4.3 Evaluation on Term Extraction

For comparison, a statistical based term candidate extraction algorithm, *TCE_SEF&CV* with the best performance (Ji and Lu, 2007), is used as a reference algorithm. Another popular algorithm which is integrated without division of steps, *TF-IDF* (Salton and McGill, 1983; Frank et al., 1999) is used as a reference method for term extraction. All the proposed algorithms and the reference algorithms need to run a term verification algorithm. For fairness, the term verification algorithm *TV_LinkA* (Term Verification – Link Analysis), is used in the second step. *TV_LinkA* is based on link analysis to calculate the relevance between the candidates and the sentences in the domain specific corpus for term verification which gives the best result in current study (a paper presenting this work is currently under review). All the algorithms rank the strings and consider the top ranked strings as term candidates. The verification of new terms is done manually.

Figure 7 shows the performance of the proposed algorithms *TCE_DI* and *TV_LinkA* for term extraction compared to the reference algorithms for IT domain using $Corpus_{IT\_Large}$. TCE_DI_{IT}, TCE_DI_{legal} and TCE_DI_{SW} indicate the proposed algorithm *TCE_DI* using different delimiter lists $DList_{IT}$, $DList_{Legal}$, and the stop word list $DList_{SW}$, respectively. As shown in Figure 7, the TCE_DI_{IT} algorithm performs best on IT domain using $DList_{IT}$. It achieves 75.4% precision when the number of extracted terms N_{TCList} reaches 5,000. The performance is 9.6% and 29.4% higher in precision compared to *TF-IDF* and *TCE_SEF&CV*, respectively. These translate to improvements of precision for over 14.8% and 63.9%, respectively.

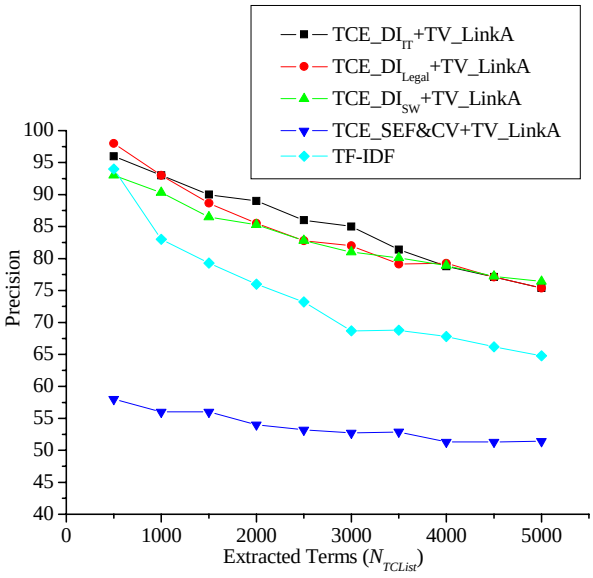

Figure 7: Performance of Different Algorithms on IT Domain

When applying the same *TV_LinkA* algorithm for term verification, *TCE_DI* using different delimiter lists provides 24% higher performance on average compared to the *TCE_SEF&CV* algorithm which translates to improvement of over 47%. The result from using delimiters of legal domain ($DList_{Legal}$) to data in IT

domain as shown in TCE_DI_{legal} is better on average than using a simple general purpose stop word list. It should be noted, however, that TCE_DI_{SW} still performs much better than the reference algorithms, which means that the proposed term candidate extraction algorithm can improve performance well even without any domain specific training.

It is also interesting to point out that the simple *TF-IDF* algorithm which was rarely used in Chinese term extraction performs better than *TCE_SEF&CV* (combined with *TV_LinkA*), which had the best performance in literature for Chinese term extraction so far. The main reason is that the test corpus consists of academic papers. Therefore, many terms are consistent and repeated a lot of times in different documents which accords with the idea of *TF-IDF*. Thus, *TF-IDF* performs relatively well because of the high-quality domain corpus. However, *TF-IDF*, as a statistics based algorithm suffers from similar problem as others statistic based methods. Thus it does not perform as well as the proposed algorithm.

Figure 8 shows the performance for the same set of algorithms for legal domain using $Corpus_{Legal\_Large}$. It can be seen that the improvement in the legal domain has similar performance and trend. The TCE_DI_{Legal} algorithm performs best on legal domain using $DList_{Legal}$. It achieves 77% precision when the number of extracted terms N_{TCList} reaches 5,000. The performance is 12.6% to 22.6% higher in precision for the 5,000 extracted terms compared to the reference algorithms which translates to improvements in precision for over 19.6% to 41.5%. The result from using delimiters of IT domain ($DList_{IT}$) to data in legal domain as shown in TCE_DI_{IT} is better on average than using a simple general purpose stop word list. This further proves that extracted delimiter list even from a different domain can be more effective than a general purpose stop word list. When applying the same *TV_LinkA* algorithm for term verification, *TCE_DI* using different delimiter lists provide 21% higher performance on average compared to the *TCE_SEF&CV* algorithm for the 5,000 extracted terms which translates to improvement of over 39%.

The performance of *TF-IDF* and *TCE_SEF&CV* are very low in the low range of N_{TCList} values compared to the counter parts in the IT domain. The main reason is that the two algorithms rely heavily on the consistency of the given corpus. However, legal articles cover much more information on different domains such are politics and economics. Thus, the reference algorithms consider some general words as terms by mistake which leads to low performance especially for the top ranked terms. The proposed *TCE_DI* algorithm achieve similar performance on legal domain compared to that on IT domain which indicates that they are less dependent on domain specific corpora.

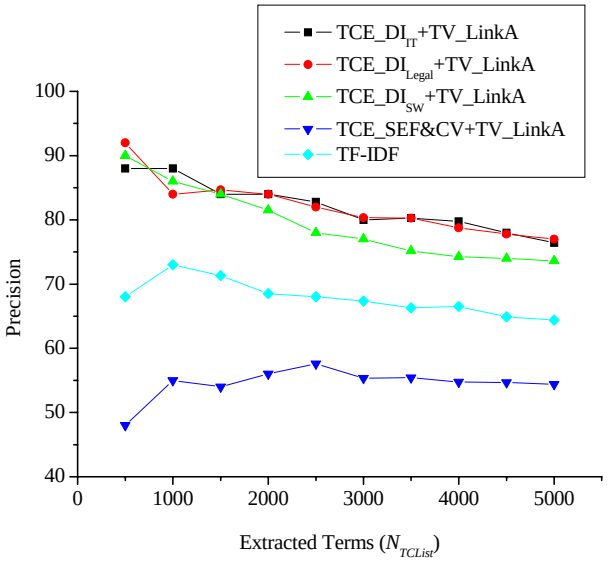

Figure 8: Performance of Different Algorithms on Legal Domain

There are three main reasons for the performance improvements of the proposed *TCE_DI* algorithm over the performance of the reference algorithms. Firstly, the delimiters which are mainly functional words (e. g. “在”(at/in), “或”(or)) and general substantive (e.g. “是”(be), “采用”(adopt)) can be extracted easily and useful to indicate term boundaries since they are quite domain independent and stable. Secondly, it is obvious that the granularity of domain specific terms in the proposed approach is much larger than that of general word segmentation. This keeps many noisy strings out of the term candidate set. Thus, the proposed delimiter based approach performs much better on term candidate selection over segmentation based statistical methods. Thirdly, the proposed approach is not as sensitive to term frequency as other statistic based approach. In the *TCE_DI* algorithm, term candidates are identified based on the identification of delimiters without regards to the frequencies of the candidates. Thus, terms having low frequencies can still be identified in the proposed approach whereas in the previous approaches including *TF-IDF*, terms with less statistical significance will be weeded out.

It is interesting to know that the proposed approach not only achieves the best performance for both domains, it also achieves second best when using delimiters extracted from a different domain. The results confirm that the delimiters are quite stable across domains and the relevance between candidates and sentences are efficient for distinguishing terms from non-terms in different domains. In fact, it also implies that the proposed approach can be applied to different domains with minimal training. In fact, if resources are limited, no training is also acceptable.

4.4 Evaluation on New Term Extraction

As there is only one ready-to-use lexicon, $Lexicon_{PKU}$ for IT domain, the evaluation on new term extraction was conducted on $Corpus_{IT\_Large}$ only. Figure 9 shows the evaluation of the proposed algorithms in terms of R_{NTE}, the ratio of new terms among all identified terms.

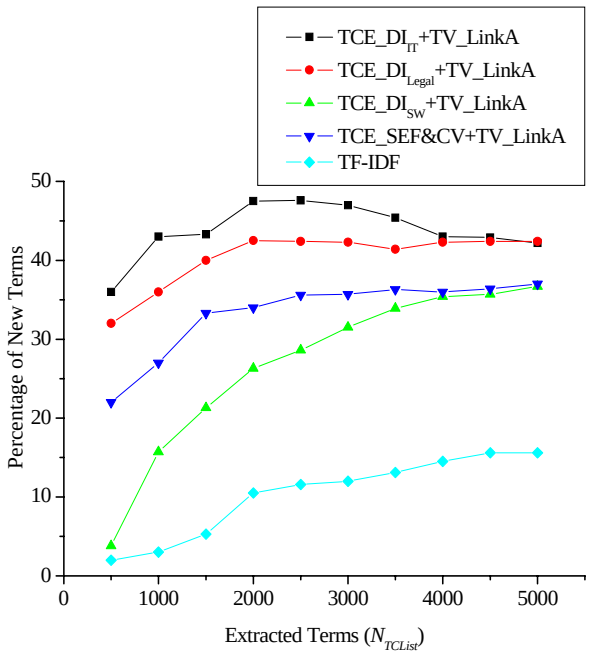

Figure 9: Performance of Different Algorithms for New Term Extraction

It can be seen that the proposed TCE_DI_{IT} algorithm is basically the top performer throughout the range. It can identify 5%(*TCE_SEF&CV*) to 27%(*TF-IDF*) more new terms compared to the reference algorithms when N_{TCList} reaches 5,000 which translates to improvements of over 15% (*TCE_SEF&CV*) to 170% (*TF-IDF*), respectively. The second best performer is TCE_DI_{legal} using delimiters of legal domain ($DList_{Legal}$). In fact, it only underperforms in the lower range of N_{TCList}. When N_{TCList} reaches 5,000, its performance is basically the same as that of TCE_DI_{IT}. However, the TCE_DI_{SW} algorithm using general purpose stop words performs much worse than using extracted delimiter lists $DList_{IT}$ and $DList_{Legal}$ as shown for TCE_DI_{IT} and TCE_DI_{legal}, respectively. In the *TCE_DI* algorithm, character strings are split by delimiters and the remained parts are taken as term candidates. Generally speaking, if a new term contains a delimiter or a stop word as its component, it cannot be identified correctly. Consequently, if a new term contains a stop word as its component, it cannot be extracted correctly using TCE_DI_{SW}. However, new terms are less likely to contain delimiters because the delimiter extraction algorithm *DList_Ext* would not consider a component as a delimiter if it is contained in a term in $Lexicon_{Domain}$. Thus, TCE_DI_{SW} picks up new terms much more slowly compared to that of TCE_DI_{IT} and TCE_DI_{legal}. Figure 9 also shows that *TF-IDF* performs the worst in new term extraction compared to other algorithms. The main reason is that new terms are not as widely used and they do not repeat a lot of times in many documents. Thus, *TF-IDF* has relatively low ability to identify new terms.

All current segmentation algorithms assume comprehensive lexical knowledge and suffer from the OOV (out of vocabulary) problem. Thus, the segmentation based term candidate extraction techniques are particularly vulnerable to new term extraction whereas the proposed approach is based on delimiters which again is more stable and domain independent. Figure 9 shows that *TCE_DI* using minimal training from different domains can extract much more new terms than previous techniques. In fact, the proposed approach can serve as a much better tool to identify new domain terms and thus be used for domain lexicon expansion.

4.5 Error Analysis

Experiments for term extraction show that the proposed algorithm in this work achieves quite significant improvements over existing algorithms. However, there is still room for improvement. Based on the analysis of experimental data, three types of errors are identified as follows.

Figure of Speech phrases. A number of long "figure of speech" phrases extracted from $Corpus_{IT\_Large}$ are considered IT terms, such as "不难看出"(it is not difficult to see that….), "新方法中"(in the new methods), "T很小时"(when T is very small), "容易证明"(it is easy to prove that….), "实验还表明"(experiments also indicate that….). These phrases are generally used in the documents as figure of speech or text patterns which often appear in academic papers or reports.

General words. A number of words from general domain extracted from $Corpus_{IT\_Large}$ are considered IT domain terms, such as "思维状态"(mental state), "声母"(initial consonant of a Chinese syllable) and "建筑"(architecture). The main reason for these errors is that these words are used in IT domain papers to describe some applications of information technology.

Long strings which contain short terms. A number of long strings which contain short terms are considered IT terms, such as "访问共享资源"(access shared resources), "再次遍历"(traverse again). Most of these errors occur because the string is made up of a short domain specific term and a general word (or character) which always occurs immediate before or after the short term and the general word is absent from the delimiter list.

Given more resources such as large domain training data, and good quality corpora of different domains for cross references, the performance of the proposed approach on the specific domain may be further improved by ameliorating these problems. However, the aim of this

study is to find a general term extraction approach using minimal resources. Thus there is a trade-off between performance and available resources.

5. Conclusion

In conclusion, this paper presents a delimiter based approach for term candidate extraction which focuses on stable and domain independent delimiters instead of looking for features associated with domain dependent terms. The proposed approach is not as sensitive to term frequency as the previous researches. It requires no prior domain knowledge, no general corpora and no adaptation for new domains. The proposed approach requires no full segmentation and considers relatively large granularity of term candidate so that many noisy strings are weeded out.

Experiments for term extraction are conducted on IT domain and legal domain, respectively. Evaluations indicate that the proposed approach has a number of advantages. Firstly, the proposed approach can improve precision of term extraction quite significantly. It achieves 14.8% to 46.7% improvements in precision over the reference algorithms for term extraction on to different domains. Secondly, the fact that the proposed approach achieves the best performance on two different domains verifies its domain independent nature. The proposed approach using delimiters extracted from a different domain also achieves the second best performance which indicates that the delimiters are quite stable and domain independent. The proposed approach still performs much better than the reference algorithms when using a general purpose stop word list, which means that the proposed approach can improve performance well even without any domain specific training. Consequently, the results demonstrate that the proposed approach can be applied to different domains without much adaptation. Thirdly, the proposed approach is particularly good for identifying new terms. It achieves 15% to 170% improvements over the current techniques for new term extraction which indicates that it can also serve as an effective tool for domain lexicon expansion.

For natural language applications, it is important to update domain knowledge and term is the most fundamental knowledge that requires continuous update. The proposed method is the best so far in automatic term extraction which can be used in a variety of NLP systems. Furthermore, the proposed approach can be applied to other related NLP tasks in the future such as in named entity extraction since these tasks are relatively similar in nature except that the delimiters may have more specific features associated with them. Even though the focus of this work is on Chinese, it would be important to know if it can be easily applied to a different language. Thus, future experiments will be conducted on different languages such as English.

6. Acknowledgements

The work described in this paper was partially supported by the Hong Kong Polytechnic University under CERG Grant B-Q941 and Central Grant: G-U297. The first author was a research assistant of the Hong Kong Polytechnic University while working on this work.

7. References

Chen, Y.R., Lu, Q., Li. W.J., Sui, Z.F., Ji, L.N. (2006). A Study on Term Extraction Based on Classified Corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, Italy, 2006.

Chien, L.F. (1999). Pat-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. *Information Processing and Management*, 35, pp.501--521.

Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G. (1999). Nevill-Manning. Domain-specific keyphrase Extraction. In Proceedings of *16th International Joint Conference on Artificial Intelligence* IJCAI-99, pp. 668--673.

Feng, H.D., Chen, K., Deng, X.T., Zheng, W.M. (2004). Accessor variety criteria for Chinese word extraction. *Computational Linguistics*. 30(1), pp.75--93.

Frantzi, K., Ananiadou, S., Mima, H. (2000) Automatic recognition of multi-word terms. *International Journal of Digital Libraries*, 3(2), pp.117--132.

Huang, C.R., Simon, P., Hsieh, S.K., Pr´evot, L. (2007). Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pp. 69–72.

Ji, L.N., Lu, Q. (2007). Chinese Term Extraction Using Window-Based Contextual Information. *CICLing 2007*, LNCS 4394, pp. 62--74.

Kageura, K., Umino, B. (1996). Methods of automatic term recognition: a review. *Term*, 3(2), pp. 259--289.

Kleinberg, J. (1997). Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pp. 668--677.

Luo, S.F., Sun, M.S. (2003). Two-Character Chinese Word Extraction Based on Hybrid of Internal and Contextual Measures. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, July, 2003, pp. 24--30.

Salton, G., McGill, M.J. (1983). Introduction to Modern Information Retrieval. McGraw-Hill.

Schone, P., Jurafsky, D. (2001). Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*, pp. 100--108.

Sornlertlamvanich, V., Potipiti, T., Charoenporn, T. (2000). Automatic corpus-based Thai word extraction with the C4.5 learning algorithm. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)* Vol. 2, Jul 2000, pp. 802--807.