Chinese Term Extraction Using Minimal Resources

Yuhang Yang
School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
1983yang@gmail.com

Qin Lu
Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
csluqin@comp.polyu.edu.hk

Tiejun Zhao
School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
tjzhao@mtlab.hit.edu.cn

Abstract

This paper presents a new approach for term extraction using minimal resources. A term candidate extraction algorithm is proposed to identify features of the relatively stable and domain independent term delimiters rather than that of the terms. For term verification, a link analysis based method is proposed to calculate the relevance between term candidates and the sentences in the domain specific corpus from which the candidates are extracted. The proposed approach requires no prior domain knowledge, no general corpora, no full segmentation and minimal adaptation for new domains. Consequently, the method can be used in any domain corpus and it is especially useful for resource-limited domains. Evaluations conducted on two different domains for Chinese term extraction show quite significant improvements over existing techniques and also verify the efficiency and relative domain independent nature of the approach. Experiments on new term extraction also indicate that the approach is quite effective for identifying new terms in a domain making it useful for domain knowledge update.

1 Introduction

Terms are the lexical units to represent the most fundamental knowledge of a domain. Term extraction is an essential task in domain knowledge acquisition which can be used for lexicon update, domain ontology construction, etc. Term extraction involves two steps. The first step extracts candidates by unithood calculation to qualify a string as a valid term. The second step verifies them through termhood measures (Kageura and Umino, 1996) to validate their domain specificity.

Existing techniques extract term candidates mainly by 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. Luo and Sun, 2003; Ji and Lu, 2007). Domain dependent features of domain terms are used in a weighted manner to identify term boundaries. However, these algorithms always face the dilemma that fewer features are not enough to identify terms from non-terms whereas more features lead to more conflicts among selected features in a specific instance.

Most term verification techniques use features on the difference in distribution of a term occurred within a domain and across domains, such as *TF-IDF* (Salton and McGill, 1983; Frank, 1999) and *Inter-Domain Entropy* (Chang, 2005). Limited distribution information on term candidates in different documents are far from enough to distinguish terms from non-terms. Other researches attempted to use more direct information. The therm verification algorithm, *TV_ConSem*, proposed in (Ji and Lu, 2007) for Chinese calculate the percentage of context words in a domain lexicon using both frequency information and semantic information. However, this technique requires a large domain lexicon and relies heavily on both the size and the quality of the lexicon. Some supervised learning

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 1033–1040
Manchester, August 2008

approaches have been applied to protein/gene name recognition (Zhou et al., 2005) and Chinese new word identification (Li et al., 2004) using *SVM* classifiers (Vapnik, 1995) which also require large domain corpora and annotations, and intensive training is needed for a new domain.

Current term extraction techniques (e.g. Frank et al., 1999; Chang, 2005; Ji and Lu, 2007) suffer from three major problems. The first problem is that these algorithms cannot identify certain kinds of terms such as the ones that have less statistical significance. The second problem is their dependency on full segmentation for Chinese text which is particularly vulnerable to handle domain specific data (Huang et al., 2007). The third problem is their dependency on some a priori domain knowledge such as a domain lexicon making it difficult to be applied to a new domain.

In this work, the proposed algorithm extracts candidates by identifying the relatively stable and domain independent term boundary markers instead of looking for features associated with the term candidate themselves. Furthermore, a novel algorithm for term verification is proposed using link analysis to calculate the relevance between term candidates and the sentences in domain specific corpus to validate their domain specificity.

The rest of the paper is organized as follows. Section 2 describes the proposed algorithms. Section 3 explains the experiments and the performance evaluation. Section 4 is the conclusion.

2 Methodology

2.1 Delimiters Based Term Candidate Extraction

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. These predecessors and successors can be considered as markers of terms, and are referred to as *term delimiters* in this paper. In contrast to terms, delimiters are relatively stable and domain independent. Thus, they can be extracted more easily. Instead of looking for features associated with terms as in other works, this paper looks for features associated with term delimiters. That is, term delimiters are identified first. Words between delimiters are then taken as term candidates.

The proposed delimiter identification based algorithm, referred to as *TCE_DI* (Term Candidate Extraction – Delimiter Identification), extracts term candidates from a domain corpus by using a delimiter list, referred to as the *DList*. Given a *DList*, the algorithm *TCE_DI* itself is straight forward. For a given character string *CS* (*CS* = $C_1C_2\ldots C_n$) shown in Figure 1, where 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 term candidates as labeled by TC_1, TC_2 and TC_3 in Figure 1, respectively. If there is no delimiter contained in *CS*, the whole string $C_1C_2\ldots C_n$ is regarded as one term candidate.

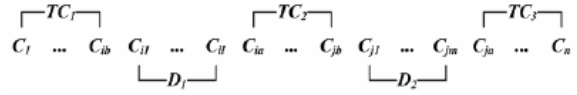

Figure 1. Paradigm of Term Candidate Extraction

DList can be obtained either from a delimiter training corpus or from a given stop word list. Given a delimiter training corpus, $Corpus_{Training}$, normally a domain specific corpus, and a domain lexicon *Lexicon*, *DList* can be obtained based on the following algorithm, referred to as *DList_Ext* (*DelimiterList Extraction Algorithm*).

Step 1: For each term T_i in *Lexicon*, mark T_i in $Corpus_{Training}$ as a non-divisible lexical unit.
Step 2: Segment remaining text in $Corpus_{Training}$.
Step 3: Extracts predecessors and successors of all T_i as delimiter candidates.
Step 4: Remove delimiter candidates that are contained in a T_i in *Lexicon*.
Step 5: Rank delimiter candidates by frequency and the top N_{DI} number of items are considered delimiters.

The *DList_Ext* algorithm basically use known terms in a domain specific *Lexicon* to find the delimiters. It can be shown in the experiments later that *Lexicon* does not need to be comprehensive. Even if a small training corpus, $Corpus_{Training}$, 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 serve as *DList* directly without using the *DList_Ext* algorithm.

2.2 Link Analysis Based Term Verification

In a domain corpus, some sentences are *domain relevant sentences* which contain more domain specific information whereas others are *general sentences* which contain less domain information. A domain specific term is more likely to be contained in domain relevant sentences, which means that domain relevant sentences and domain specific terms have a mutually reinforcing relationship. A novel algorithm, referred to as *TV_LinkA* (Term Verification – Link Analysis) based the Hyperlink-Induced Topic Search (*HITS*) algorithm (Kleinberg, 1997) originally proposed for information retrieval, is proposed using link analysis to calculate the relevance between term candidates and the sentences in domain specific corpora for term verification.

In *TV_LinkA*, a node *p* can either be a sentence or a term candidate. If a term candidate $Term_C$ is contained in a sentence *Sen* of the corpus $Corpus_{Extract}$ where the candidates were extracted, there is a directional link from *Sen* to $Term_C$. This way, a graph for the candidates and the sentences in $Corpus_{Extract}$ can be constructed and the links between them indicate their relationships. A good *hub* in $Corpus_{Extract}$ is a sentence that contains many good authorities; a good *authority* is a term candidate that is contained in many good hubs. Each node *p* is associated with a non-negative authority weight $w(p)^A$ and a non-negative hub weight $w(p)^H$. Link analysis in *TV_LinkA* makes use of the relationship between hubs and authorities via an iterative process to maintain and update authority/hub weights for each node of the graph.

Let V^A denote the authority vector $(w(p_1)^A, w(p_2)^A,\ldots, w(p_n)^A)$ and V^H denote the hub vector $(w(p_1)^H, w(p_2)^H,\ldots, w(p_n)^H)$, where *n* is the sum of the total number of sentences and the total number of term candidates. Given weights V^A and V^H with a directional link $p \rightarrow q$, the *I* operation(an in-pointer to a node) and the *O* operation(an out-pointer to a node) update $w(q)^A$ and $w(p)^H$ as follows.

$$I \text{ operation: } w(q)^A = \sum_{p \rightarrow q \in E} w(p)^H \qquad (1)$$

$$O \text{ operation: } w(p)^H = \sum_{p \rightarrow q \in E} w(q)^A \qquad (2)$$

Let *k* be the iteration termination parameter and *z* be the vector (1, 1, 1,…, 1) , and V^A and V^H are initialized to $V_0^A = V_0^H = z$. Hubs and authorities can then be calculated as follows.

For $i = 1, 2,\ldots, k$

Apply the *I* operation to (V_{i-1}^A , V_{i-1}^H), obtaining new V'^A_i .

Apply the *O* operation to (V'^A_i , V_{i-1}^H), obtaining new V'^H_i .

Normalize V'^A_i by dividing the normalization factor $\sqrt{\sum (w'(p)^A)^2}$ to obtain V_i^A.

Normalize V'^H_i by dividing the normalization factor $\sqrt{\sum (w'(p)^H)^2}$ to obtain V_i^H .

End

Return (V_k^A , V_k^H)

In $Corpus_{Extract}$, term candidates with high authority in a few documents are likely to be domain specific terms whereas candidates with high authority in many documents are more likely to be commonly used general words. Based on this observation, the termhood of each candidate term $Term_C$, denoted as $Termhood_C$, is calculated according to formula (3) defined below.

$$Termhood_C = (\sum_j w(C)_j^A) \log(\frac{|D|}{DF_C}) \qquad (3)$$

where $w(C)_j^A$ is the authority of $Term_C$ in a document D_j of $Corpus_{Extract}$, $|D|$ is the total number of documents in $Corpus_{Extract}$ and DF_C is the total number of documents in which $Term_C$ occurs. $Term_C$s are then ranked according to their termhood values $Termhood_C$, and the top ranked N_{TCList} candidates are considered terms. N_{TCList} is an algorithm parameter to be determined experimentally.

3 Performance Evaluation

3.1 Data Preparation

To evaluate the performance of the proposed algorithms for Chinese, experiments are conducted on four corpora of two different domains as listed in Table 1. $Corpus_{IT\_Small}$ and $Corpus_{IT\_Large}$ are two sets of non-overlapping academic papers in the IT domain and $Corpus_{IT\_Small}$ is identical to the corpus used in *TV_ConSem(Ji and Lu, 2007)*. $Corpus_{Legal\_Small}$ is a complete set of official Chinese criminal law articles. $Corpus_{Legal\_Large}$ includes the complete set

of official Chinese constitutional law articles and Economics/Finance law articles (http://www.law-lib.com/). Three domain lexicons used in the experiments are detailed in Table 2. $Lexicon_{IT}$ is obtained according to the term extraction algorithm (Ji and Lu, 2007) with manual verification. $Lexicon_{Legal}$ is extracted from $Corpus_{Legal\_Small}$ by manual verification too. Because legal text covers a lot of different areas such finance, science, advertisement, etc., the actually legal specific terms are relatively small in size. $Lexicon_{PKU}$ contains a total of 144K manually verified IT terms supplied by the Institute of Computational Linguistics, Peking University. $Lexicon_{PKU}$, is used as the standard term set for evaluation on the IT domain. $Corpus_{IT\_Small}$ and $Lexicon_{IT}$ are used to obtain the delimiter list of IT domain, $DList_{IT}$. $Corpus_{Legal\_Small}$ and $Lexicon_{Legal}$ are used to obtain the delimiter list of legal domain, $DList_{Legal}$. $Corpus_{IT\_Large}$ and $Corpus_{Legal\_Large}$ are used as open test data to evaluate the proposed algorithms in IT domain and legal domain, respectively.

Corpus	Domain	Size	Text type
$Corpus_{IT\_Small}$	IT	77K	Academic papers
$Corpus_{IT\_Large}$	IT	6.64M	Academic papers
$Corpus_{Legal\_Small}$	Legal	344K	Law article
$Corpus_{Legal\_Large}$	Legal	1.04M	Law article

Table 1. Different Corpora Used for Experiments

Lexicon	Domain	Size	Source
$Lexicon_{IT}$	IT	3,337	$Corpus_{IT\_Small}$
$Lexicon_{Legal}$	Legal	394	$Corpus_{Legal\_Small}$
$Lexicon_{PKU}$	IT	144K	PKU

Table 2. Different Lexicons Used for Experiments

To verify that the approach works with a simple stop word list without delimiter extraction, a stop word list, $DList_{SW}$, is also used as reference by taking the 494 general purpose stop words downloaded from a Chinese NLP resource website (www.nlp.org.cn) without any modification.

The performance of the algorithm in the IT domain is evaluated by precision according to the follow formula:

$$precision_{TE} = \frac{N_{Lexicon} + N_{New}}{N_{TCList}} \quad (4)$$

where N_{TCList} is the number of term candidates in term candidate list *TCList* extracted by an evaluated algorithm, $N_{Lexicon}$ denotes the number of term candidates in *TCList* contained in $Lexicon_{PKU}$, N_{New} denotes the number of extracted term candidates that are not in $Lexicon_{PKU}$, yet are considered correct. Thus, N_{New} is the number of newly discovered terms with respect to $Lexicon_{PKU}$. The verification of all the new terms is carried out manually by two experts independently. A new term is considered correct if both experts marked them as correct terms. As there is no reasonably large standard legal term list available, the evaluation of the legal domain in terms of precision is conducted manually. No evaluation on new term extraction is conducted.

To evaluate the ability of the algorithms in identify new terms in the IT domain, another measurement is applied to the IT corpus against $Lexicon_{PKU}$ based on the following formula:

$$R_{NTE} = \frac{N_{New}}{N_{TCList}} \quad (5)$$

where *TCList* and N_{New} are the same as given in formula (4). A higher R_{NTE} indicates that more extracted terms are outside of $Lexicon_{PKU}$ and are thus considered new terms. This is similar to the measurements of out of vocabulary (OOV) in Chinese segmentation. A higher R_{NTE} indicates the algorithm can be useful for domain knowledge update including lexicon expansion.

3.2 Evaluation on Term Extraction

For comparison, a statistical based term candidate extraction algorithm, *TCE_SEF&CV* with the best performance in (Ji and Lu, 2007) using both internal association and external strength, is used as the reference algorithm for the evaluation of *TCE_DI*. A statistics based term verification algorithm, *TV_ConSem* (Ji and Lu, 2007) using semantic information within a context window is used for the evaluation of *TV_LinkA*. $Lexicon_{PKU}$ is also used in *TV_ConSem*. Two popular methods integrated without division of candidate extraction and verification steps are used for comparison. The first one is based on *TF-IDF* (Salton and McGill, 1983; Frank et al., 1999). The second one is a supervised learning approach based on a *SVM* classifier, SVM^{light} (Joachims, 1999). The features used by SVM^{light} are shown in Table 3. Two training sets are constructed for the *SVM* classifier. The first one includes 3,337 positive examples ($Lexicon_{IT}$) and 5,950 negative examples extracted from $Corpus_{IT\_Small}$. The second one includes 394 positive examples

($Lexicon_{Legal}$) and 28,051 negative examples extracted from $Corpus_{Legal\_Small}$.

No.	Feature Explanation
1	Percentage of the Chinese characters occurred in $Lexicon_{Domain}$
2	Frequency in the domain corpus
3	Frequency in the general corpus
4	Part of speech
5	The length of Chinese characters in the candidate
6	The length of non-Chinese characters in the candidate
7	Contextual evidence

Table 3. Features Used in the SVM Classifier

Figure 2 shows the performance of the proposed *TCE_DI* and *TV_LinkA* for term extraction compared to the reference algorithms for IT domain using $Corpus_{IT\_Large}$. TCE_DI_{IT} and TCE_DI_{legal} indicate *TCE_DI* using extracted delimiter lists $DList_{IT}$ and $DList_{Legal}$ with N_{DI} = 500, respectively. TCE_DI_{SW} simply uses the stop word list $DList_{SW}$.

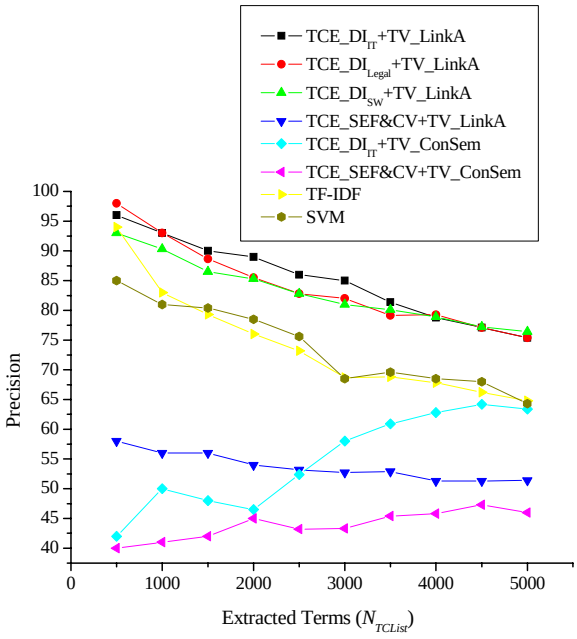

Figure 2 Performance of Different Algorithms on IT Domain

As shown in Figure 2, term extraction based on TCE_DI_{IT} combined with *TV_LinkA* gives the best performance. 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* combined with *TV_ConSem*, respectively. These translate to improvements of precision of over 14.8% and 63.9%, respectively.

When applying the same *TV_LinkA* algorithm for term verification, *TCE_DI* using different delimiter lists provide 24% better 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 stop word list. It should be noted, however, that TCE_DI_{SW} still performs much better than the reference algorithms, which means that delimiter based term candidate extraction algorithm can improve performance even without any domain specific training. When applying the same TCE_DI_{IT} algorithm in term candidate extraction, *TV_LinkA* provides 10% higher performance compared to the *TV_ConSem* algorithm which translates to improvement of over 15.3%. It is important to point out that *TV_LinkA* using only the stop word list without any domain specific knowledge performs better than *TV_ConSem* using a large domain lexicon. In other words, delimiter based extraction with link analysis use much less resources and still improve performance of *TV_ConSem*.

The performance of TCE_DI_{IT} or *TCE_SEF&CV* combined with *TV_ConSem* have an upward trend when more terms are extracted which seems to be against intuition. The principle of the *TV_ConSem* algorithm is that a candidate is considered a valid term if a majority of its context words already appear in the domain lexicon. General words are more likely to be ranked on top because they are commonly used which explains the low performance of *TV_ConSem* in the lower range of N_{TCList}. When N_{TCList} increases, more domain terms are included. Thus, there is an upward trend in precision. But, the upward trend reverts at around 4,500 because the measurement in percentage is too low to distinguish valid terms from non-term candidates.

It is also interesting to point out that the simple *TF-IDF* algorithm which was rarely used in Chinese term extraction performs as well as the *SVM* classifier. The main reason is that the test corpus consists of academic papers. So, 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 based on

statistics. Thus it does not perform as well as the proposed *TCE_DI* and *TV_LinkA* algorithms.

Figure 3 shows that the proposed algorithms achieve similar performance on the legal domain. TCE_DI_{Legal} combined with *TV_LinkA* perform the best. The result from using IT domain delimiters ($DList_{IT}$) in legal domain as shown in TCE_DI_{IT} is better on average than using the general purpose stop list. This further proves that extracted delimiter list even from a different domain can be more effective than a general stop word list. When applying the same *TV_LinkA* algorithm for term verification, *TCE_DI* using different delimiter lists are better than all the reference algorithms. Without large lexicon in Chinese legal domain, the *TV_ConSem* algorithm does not even work. *TV_LinkA* using no prior domain knowledge for term verification still achieves similar improvement compared to that of the IT domain where a comprehensive domain lexicon is available.

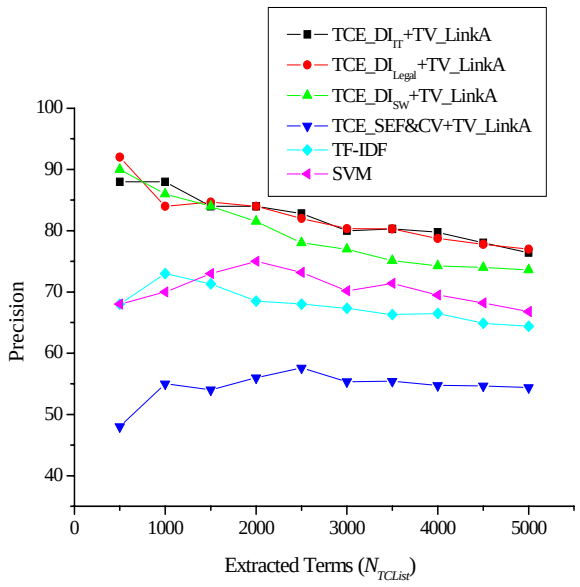

Figure 3. Performance of Different Algorithms on Legal Domain

There are three main reasons for the performance improvements of the proposed *TCE_DI* and *TV_LinkA* 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 are effective term boundary markers since they are quite domain independent and stable. Secondly, the granularity of domain specific terms extracted the proposed algorithm is much larger than words obtained by word segmentation. This keeps many noisy strings out of the term candidate set. Thus, the proposed delimiter based algorithm performs much better over segmentation based statistical methods. Thirdly, the proposed approach is not as sensitive to term frequency as other statistical based approaches because term candidates are identified without regards to the frequencies of the candidates. In the *TV_LinkA* algorithm, terms are verified by calculating the relevance between candidates and the sentences instead of the distributions of terms in different types of documents. Terms having low frequencies can be identified as long as they are in domain relevant sentences whereas in the previous approaches including *TF-IDF*, terms with less statistical significance are weeded out. For example, a long IT term “层次化存储系统” (Hierarchical storage system) with a low frequency of 6 is extracted using the proposed approach. It cannot be identified by *TF-IDF* since the statistical information is not significant. This term cannot be extracted by the segmentation based algorithms either because general segmentor split long terms into pieces making them difficult to be reunited using term extraction techniques.

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 extracted delimiters from a different domain. The results confirm that 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, the proposed approach can be applied to different domains with minimal training or no training if resources are limited.

3.3 Evaluation on New Term Extraction

As $Lexicon_{PKU}$ is the only ready-to-use domain lexicon, the evaluation on new term extraction is conducted on $Corpus_{IT\_Large}$ only. Figure 4 shows the evaluation of the proposed algorithms compared to the reference algorithms in terms of R_{NTE}, the ratio of new terms among all identified terms.

It can be seen that the proposed algorithms TCE_DI_{IT} combined with *TV_LinkA* is basically the top performer throughout the range. It can identify 4% (with respect to *TCE_SEF&CV +TV_ConSem*) to 27% (with respect to *TF-IDF*) more new terms when N_{TCList} reaches 5,000 which translate to improvements of over 9% to 170%, respectively. The second best performer is TCE_DI_{legal} combined with *TV_LinkA* using delimiters of legal domain. In fact, it only underperforms in the lower range of N_{TCList}

compared to TCE_DI_{IT}. When N_{TCList} reaches 5,000, their performance is basically the same. However, the TCE_DI_{SW} algorithm using stop words performs much worse than using extracted delimiter lists as shown for TCE_DI_{IT} and TCE_DI_{legal}. 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 $Lexicon_{Domain}$. Consequently, TCE_DI_{SW} is less adaptive to domain specific data compared to TCE_DI_{IT} and TCE_DI_{legal}. That is also why TCE_DI_{SW} picks up new terms much more slowly.

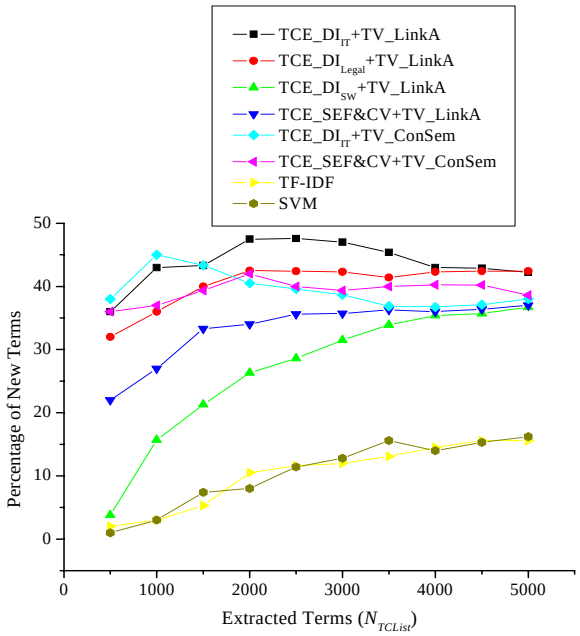

Figure 4. Performance of Different Algorithms for New Term Extraction

It is interesting to know that TCE_DI_{IT} combined with *TV_ConSem* identifies more new terms in the low range of N_{TCList}. In the *TV_ConSem* algorithm, the major information used for term verification is the percentage of the context words appear in the domain lexicon. As discussed earlier in Section 3.2, *TV_ConSem* ranks commonly used general words higher than others which leads to the low precision of *TV_ConSem* for term extraction. A new term faces a similar scenario because more of its context words occur in the domain lexicon than that of other terms. Thus, new terms are actually ranked higher than other terms in *TV_ConSem* which explains its higher ability to identify new terms in the low range of N_{TCList}. However, its performance drops in the high range of N_{TCList} because the influence of context words diminishes in terms of percentage in the domain lexicon to distinguish terms from non-terms. Figure 4 also shows that *TF-IDF* and *SVM* perform the worst in new term extraction compared to other algorithms. *TF-IDF* has relatively low ability to identify new terms since new terms are not widely used and they do not repeat a lot of times in many documents. As *SVM* is sensitive to training data, it is naturally not adaptive to new terms.

All current Chinese term extraction algorithms rely on segmentation with comprehensive lexical knowledge and yet Chinese segmentation algorithms have the OOV (out of vocabulary) problem. This makes Chinese term extraction particularly vulnerable to new term extraction. The proposed approach, on the other hand, is based on delimiters which is more stable, domain independent, and OOV independent. Figure 4 shows that *TCE_DI* and *TV_LinkA* 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 can be quite effective for domain lexicon expansion.

4 Conclusion

In conclusion, this paper presents a robust term extraction approach using minimal resources. It includes a delimiter based algorithm for term candidate extraction and a link analysis based algorithm for term verification. The proposed approach is not sensitive to term frequency as the previous works. It requires no prior domain knowledge, no general corpora, no full segmentation, and minimal adaptation for new domains.

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. 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 as a completely unsupervised approach without any training. Consequently, the results demonstrate that the proposed approach can be applied to different domains easily even without training. Thirdly, the proposed approach is particularly good for identifying new terms so that it can serve as an effective tool for domain lexicon expansion.

Acknowledgements

This work was done while the first author was working at the Hong Kong Polytechnic University supported by CERG Grant B-Q941 and Central Research Grant: G-U297.

References

Chang Jing-Shin. 2005. Domain Specific Word Extraction from Hierarchical Web Documents: A First Step toward Building Lexicon Trees from Web Corpora. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Learning*: 64-71.

Chien LF. 1999. Pat-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. *Information Processing and Management*, vol.35: 501-521.

Eibe Frank, Gordon. W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific Keyphrase Extraction. In *Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99*: 668-673.

Feng Haodi, Kang Chen, Xiaotie Deng , and Weimin Zheng, 2004. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75-93.

Hiroshi Nakagawa, and Tatsunori Mori. 2002. A simple but powerful automatic term extraction method. In *COMPUTERM-2002 Proceedings of the 2nd International Workshop on Computational Term*: 29-35. Taiwan, August 2002.

Hisamitsu T., and Y. Niwa. 2002. A measure of term representativeness based on the number of co-occurring salient words. In *Proceedings of the 19th COLING, 2002*.

Huang Chu-Ren, Petr ˇSimon, Shu-Kai Hsieh, and Laurent Pr´evot. 2007. Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification. In *Proceedings of the ACL 2007 Demo and Poster Sessions*: 69–72. Joachims T. 2000. Estimating the Generalization Performance of a SVM Efficiently. In *Proceedings of the International Conference on Machine Learning*, Morgan Kaufman, 2000.

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

Kleinberg J. 1997. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*: 668-677. New Orleans, America, January 1997.

Ji Luning, and Qin Lu. 2007. Chinese Term Extraction Using Window-Based Contextual Information. In *Proceedings of CICLing 2007, LNCS 4394*: 62 – 74.

Li Hongqiao, Chang-Ning Huang, Jianfeng Gao, and Xiaozhong Fan. The Use of SVM for Chinese New Word Identification. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNL P2004)*: 723-732. Hainan Island, China, March 2004.

Luo Shengfen, and Maosong Sun. 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*: 24-30.

McDonald, David D. 1993. Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text, pages 32--43, Columbus, OH, June. Special Interest Group on the Lexicon of the Association for Computational Linguistics.

Nasreen AbdulJaleel and Yan Qu. 2005. Domain Term Extraction and Structuring via Link Analysis. In *Proceedings of the AAAI '05 Workshop on Link Analysis*: 39-46.

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

Schone, P. and Jurafsky D. 2001. Is Knowledge-free Induction of Multiword Unit Dictionary Headwords a solved problem? In *Proceedings of EMNLP2001*.

Sornlertlamvanich V., Potipiti T., and Charoenporn T. 2000. Automatic Corpus-based Thai Word Extraction with the C4.5 Learning Algorithm. In *Proceedings of COLING 2000*.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, 1995.

Zhou GD, Shen D, Zhang J, Su J, and Tan SH. 2005. Recognition of Protein/Gene Names from Text using an Ensemble of Classifiers. *BMC Bioinformatics* 2005, 6(Suppl 1)**:**S7.