Fault-Tolerant Learning for Term Extraction ∗

Yuhang Yang, Hao Yu, Yao Meng, Yingliang Lu and Yingju Xia

Fujitsu Research & Development Center Co., LTD.
15/F, Tower A, Ocean International Center,
No.56 Dong Si Huan Zhong Rd, Chaoyang District, Beijing, 100025, P.R. China
{yyh, yu, mengyao, luyl, yjxia}@ cn.fujitsu.com

Abstract. This paper presents the Fault-Tolerant Learning approach for term extraction. The approach extracts terms using automatically generated seeds instead of prior domain knowledge or annotated corpora. Thus it is applicable to any domain specific corpus and it is especially useful for resource-limited domains. Two classifiers are separately trained for prediction and verification to ensure the performance of the proposed approach. Evaluations conducted on two different domains for Chinese term extraction show significant improvements over existing techniques and also verify the efficiency and relative domain independent nature of the approach.

Keywords: Fault-Tolerant Learning, term extraction, machine learning.

1 Introduction

Terms are the lexical units to represent the most fundamental knowledge of a domain. Term extraction aims to extract meaningful words or phrases representing domain specific meaning or concepts. Thus two issues are considered in term extraction. The first issue is to identify boundaries of meaningful words and phrases. The second issue is to verify terms by calculating domain specificity (Kageura and Umino, 1996).

Existing term extraction techniques can be divided into four main categories including statistics based measures, trigger words (or characters) based algorithms, domain knowledge based methods and supervised methods.

The first category is statistics based measures which identify terms by their statistical significance. The most widely used statistical measurement is *TF-IDF* (Salton and McGill, 1983; Frank, 1999), which is based on the hypothesis that “if a candidate occurs frequently in a few documents of a domain, it is likely a term”. The co-occurrences between the target string and its components or context, referred to as *Internal association* (e.g. Schone and Jurafsky, 2001) and *context dependency* (e.g. Sornlertlamvanich et al., 2000), are used for term extraction. There are also studies evaluating the distribution of a term within a domain or across domains through different metrics, such as term representativeness (Hisamitsu and Niwa, 2002), *Inter-Domain Entropy* (Chang, 2005) and the *Lexicon Set Algorithm* (Chen et al., 2006). Statistics based techniques can extract common used terms with statistical significance. However, the techniques are very sensitive to term frequency, and thus terms with low frequencies cannot be extracted.

The second category is based on trigger words or characters. According to (Feng et al., 2004) and (Yang et al., 2008), characters and words immediately before and after these terms are proven to be useful for term extraction. *Accessor Variety Criteria* proposed in (Feng et al., 2004) considers the characters that are directly before or after a string as important factors for determining the independence of the string. TE_{Del} (delimiter based term extraction) proposed in

∗ Copyright 2010 by Yuhang Yang, Hao Yu, Yao Meng, Yingliang Lu and Yingju Xia

(Yang et al., 2008) identifies terms by finding their predecessors and successors as term boundary markers. Strings between delimiters are taken to be term candidates.

The third category is based on some a priori domain knowledge such as a large domain lexicon. Nakagawa (2002) identified compound nouns as domain specific terms by measuring the domain specificity of the component words, which is determined by finding out whether they appear in the domain lexicon. But this method cannot deal with non-compound terms. TE_{Kno} (knowledge based term extraction), proposed in (Ji and Lu, 2007) for Chinese, calculates the percentage of context words in a domain lexicon using both frequency information and semantic information. TE_{Kno} also requires an existing domain lexicon for verification.

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

As described before, different categories of term extraction techniques suffer from different problems. Statistics-based methods cannot identify terms without statistical significance since they are very sensitive to term frequency. Trigger word based algorithms, which use only limited features, are likely to extract certain kinds of terms but miss the others. Knowledge based algorithms and supervised methods rely heavily on both the size and the quality of domain knowledge or annotated training data which makes it difficult to be applied to a new domain.

In this work, the *Fault-Tolerant Learning* (*FTL*) approach is proposed to overcome these problems. After automatically generating two sets of seeds based on two unsupervised algorithms, different classifiers are separately trained using different seed sets, followed by double checking for term verification. The proposed *FTL* approach extracts terms using automatically generated seeds instead of domain knowledge or annotated corpora. Thus it is applicable to any domain specific corpus. It is especially useful for knowledge-limited and resource-limited domains. Two classifiers are separately trained for prediction and verification which aims to improve the performance. Moreover, all the features are used in each classifier which makes it possible to cover more kinds of terms.

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 Overview of Fault-Tolerant Learning

Fault-Tolerant (Laprie, 1985) is the property that provides, by redundancy, service complying with the specification in spite of faults having occurred or occurring.

Fault-Tolerant Learning proposed in this study makes use of seeds produced by unsupervised techniques without manual checking. Thus noise may exist from the beginning and could increase after each iteration. Two classifiers separately trained are used for prediction and verification which aims to make the results more reliable by handling the noise. That is why we call the proposed approach *Fault-Tolerant Learning*. *FTL* is based on two assumptions. First, the accuracy of the automatically generated seeds is higher than a random selection mechanism. This insures that the generated seeds are useful. Second, two conditional independent views can be obtained to make the verification more efficient.

The proposed *FTL* approach is inspired by *Transfer learning* and *Co-training*. The study of *Transfer learning* (Ando and Zhang, 2005) is motivated by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions. *Transfer learning* allows the domains, tasks, and distributions used in training and testing to be different. Both *Transfer learning* and *FTL* focus on tasks without enough labeled training data. *Transfer learning* uses resources from different domains or tasks for transferring knowledge to the target domain or task. *FTL* uses noisy instances automatically labeled by unsupervised algorithms instead. *Co-training* (Blum and Mitchell, 1998) is a representive bootstrapping method, which starts with a set of labeled data, and increase the amount of annotated data using some amounts of unlabeled data in an incremental way. Both *Co-training* and *FTL* require two independent views for verification. The main difference between them is that *FTL* starts with noisy seeds instead of manually checked seeds in *Co-training*. Besides, *Co-training* always uses two split feature spaces whereas *FTL* relaxes the definition of two independent views. In this study, two sets of seeds produced by different algorithms are taken as two different views. The main reason is that a few features may be useful only for certain kinds of terms which has been described in Section 1.

The framework of *Fault-Tolerant Learning* is shown in Figure 1. It consists of 5 steps listed below.

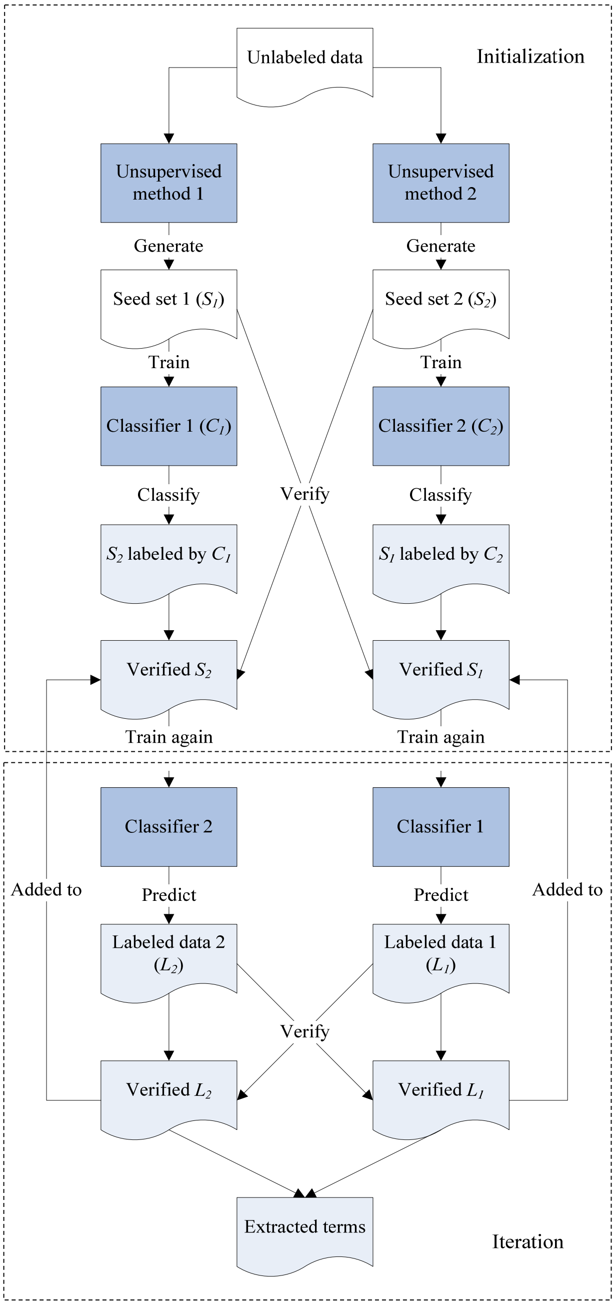

Figure 1: Framework of Fault-Tolerant Learning.

Step 1: Given an unlabeled data set, two sets of seeds are extracted using two unsupervised methods.

Step 2: Two classifiers are trained by using different sets of seeds, respectively.

Step 3: One classifier trained by one seed set is applied to verify the other set of seeds.

Step 4: The classifiers are trained again using the verified seed sets.

Step 5: Two classifiers are used for predicting and verification, instances with the most confidence are considered correct and added into the seed sets.

The *FTL* approach works with unlabeled data by making use of noisy seeds. There are two crucial issues in the proposed model. The first issue is to handle noise. From the beginning to each iteration, two classifiers are used for prediction and verification, respectively. Different types of feature are integrated in each classifier to make it more efficient. The second issue is to maintain two independent views. The independent views are obtained by using two different seed sets in this study. At the beginning, two seed sets are produced using different algorithms. In each iteration, two sets of new labeled instances are extracted and added to different seed sets respectively.

2.2 Unsupervised Algorithms

For term extraction, *TF-IDF* and the delimiter based algorithm (referred to as TE_{Del}) are selected as unsupervised algorithms based on two reasons. First, *TF-IDF* and TE_{Del} are proven to perform well for term extraction, especially when the number of extracted terms is small. Second, they are based on different features. *TF-IDF* is based on statistical significance whereas TE_{Del} is based on trigger words for term boundary detection. Thus different kinds of terms can be extracted which satisfies the condition of two independent views required in *FTL*.

TF-IDF (Salton and McGill, 1983) is the representive statistical measure which calculates the distribution of a string in different documents.

$$TFIDF(TC_i) = TF(TC_i) \cdot IDF(TC_i) \tag{1}$$

$$IDF(TC_i) = \log(\frac{|D|}{DF(TC_i)}) \tag{2}$$

where $TF(TC_i)$ is the number of times term candidate TC_i occurs in the domain corpus, $DF(TC_i)$ is the number of documents in which TC_i occurs at least once, $|D|$ is the total number of documents in the corpus, $IDF(TC_i)$ is the inverse document frequency which can be calculated from the document frequency.

TE_{Del} (Yang et al., 2008; Yang et al., 2009) identifies the relatively stable and domain-independent delimiter words immediately before and after domain specific terms for term candidate extraction. Delimiters are likely to be either functional words or other general substantives connecting terms and are proven to be useful for term boundary identification. The method verifies terms by using different types of relevance including candidate-candidate relevance, candidate-sentence relevance and candidate-document relevance.

2.3 Fault-Tolerant Learning for Term Extraction

The proposed approach extracts terms from a raw domain specific corpus $Corpus_D$. In *FTL*, a basic classification algorithm is required to construct C_1 and C_2. The widely used *SVM* classifier is adopted as the basic classifier in this study. The details of the *FTL* based term extraction approach are shown in Figure 2.

```
Given:
    -   A raw domain specific corpus $Corpus_D$;
    -   A set of unlabeled strings $U$;
Initialization:
    1.  Generate the first seed set $S_1$ using $TF\text{-}IDF$;
    2.  Generate the second seed set $S_2$ using $TE_{Del}$;
    3.  Learn the first classifier $C_1$ based on $S_1$;
    4.  Learn the second classifier $C_2$ based on $S_2$;
    5.  Use $C_1$ to label automatically generated seed set $S_2$;
    6.  Use $C_2$ to label automatically generated seed set $S_1$;
    7.  Remove instances with different tags labeled by $C_1$ and $C_2$ from $S_1$ and $S_2$;
Loop for $I$ iterations:
    8.  Learn $C_1$ based on $S_1$;
    9.  Learn $C_2$ based on $S_2$;
    10. Use $C_1$ to label instances from $U$;
    11. Produce $L_1$ consisting of $p$ terms and $n$ non-terms with the most confidence from $U$;
    12. Use $C_2$ to label instances from $L_1$;
    13. Remove instances with different tags labeled by $C_1$ and $C_2$ from $L_1$;
    14. Add instances from $L_1$ to $S_1$;
    15. Use $C_2$ to label instances from $U$;
    16. Produce $L_2$ consisting of $p$ terms and $n$ non-terms with the most confidence from $U$;
    17. Use $C_1$ to label instances from $L_2$;
    18. Remove instances with different tags labeled by $C_1$ and $C_2$ from $L_2$;
    19. Add instances from $L_2$ to $S_2$;
```

Figure 2: The FTL based term extraction approach

2.4 Used Features

Different types of features are useful for term extraction according to previous studies. However, most existing techniques use only a few features. Thus only certain kinds of terms can be extracted. Four types of features shown as follows are integrated in the proposed approach.

Frequency: Frequency is the most fundamental feature for term extraction. Most statistical measures (Luo and Sun, 2003) are based on term frequency, document frequency and frequencies of its components or context. The frequencies of each unlabeled instance and its components are counted.

Part Of Speech (POS): POS is a linguistic category of words generally defined by the syntactic or morphological behaviour of the lexical item. Many domain specific terms are descriptive and very long. Therefore, POS may provide useful evidence about the term boundaries.

Delimiter: Delimiters are relatively stable and domain-independent which occurs immediately before and after domain specific terms. They are useful for term boundary identification. Delimiters are distinguished as predecessors and successors by their positions instead of being equally considered in (Yang et al., 2008).

Head/End Word: It has been well-documented that the head/end word features are very useful. Candidates with certain head words or end words are more likely to be certain kinds of terms. Thus the statistical information of head words and end words are collected to distinguish terms from non-terms.

3 Performance Evaluation

3.1 Data Preparation

Most of the existing term extraction algorithms are conducted on the IT domain (Frank et al., 1999; Nakagawa and Mori, 2002; Ji and Lu, 2007), although some are done in other domains. To validate the relative independence of the proposed method on certain domains, experiments are conducted on two basically unrelated domains, namely, the IT domain and the legal domain. The data in each domain is split into two non-overlapping sets for training and testing. The four corpora of IT domain and legal are listed in Table 1 showing their sizes and sources.

Table 1: Different corpora used for experiments.

Corpus	Domain	Size	Text type
$Corpus_{IT\_Train}$	IT	77K	Academic papers
$Corpus_{IT\_Test}$	IT	6.64M	Academic papers
$Corpus_{Legal\_ Train}$	Legal	344K	Law Article
$Corpus_{Legal\_Test}$	Legal	1.04M	Law Article

Table 2 shows the three domain lexicons used in some reference algorithms. $Lexicon_{IT}$ and $Lexicon_{Legal}$ are manually verified from $Corpus_{IT\_Train}$ and $Corpus_{Legal\_Train}$, respectively. $Lexicon_{PKU}$ contains a total of 144K manually verified IT terms supplied by the Institute of Computational Linguistics, Peking University. All the three domain lexicons are used in some reference algorithms.

Table 2: Different lexicons used for experiments

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

As described before, $Corpus_{IT\_Test}$ and $Corpus_{Legal\_Test}$ are used for test. The other two domain corpora and all the three domain lexicons are used as training data or prior domain knowledge in some reference algorithms. The details will be given in Section 3.3. It should be pointed out that the proposed approach works based on a raw domain corpus without any domain knowledge or annotated training data.

3.2 Evaluation Metric

Performance is mainly measured by precision. As precision is measured with respect to the number of extracted terms, recall is indirectly measured. Due to the large number of extraction results, random sampling is used to select the data for manual verification. Basically, one term is selected for every 10 extracted terms. To avoid any bias towards a particular method, all of the sampled data from different algorithms are first scrambled, and then independently evaluated by two persons. If there is a discrepancy between the two evaluators, another review is conducted.

3.3 Baseline Methods

For comparison, four algorithms having good performance in literature are taken as baseline methods. The baseline methods consist of a statistical based algorithm *TF-IDF*, a delimiter based algorithm TE_{Del}, a prior knowledge based method TE_{Kno} (Ji and Lu, 2007) and a supervised learning method *SVM*. *TF-IDF* and TE_{Del} have been described in Section 2.2. The main ideas of TE_{Kno} and *SVM* are shown as follows.

TE_{Kno} extracts term candidates using both internal association and external strength, and uses semantic information within a context window for term verification. TE_{Kno} verifies a candidate as a term if the percentage of its contextual words found in an existing domain lexicon is higher than a predefined threshold. $Lexicon_{PKU}$ is used as an existing lexicon of IT domain in TE_{Kno}.

SVM classifier (Vapnik, 1995) is a typical supervised learning approach which has been widely used in many NLP tasks, such as text classification. *SVM* uses all the features listed in Section 2.4 for comparison with the proposed approach. Two training sets, generated using *TF-IDF* and TE_{Del} with manual verification, are constructed for the *SVM* classifier. The first one includes 595 positive examples and 942 negative examples extracted from IT domain. The second one includes 636 positive examples and 957 negative examples extracted from legal domain.

In the TE_{Del} algorithm, a delimiter list *DList* is required for term extraction. *DList* can be obtained either from a delimiter training corpus or from a given stop word list. *DList* obtained from training data is proved to perform better than a stop word list. Thus training data are used to guarantee the best performance of the reference algorithm. $Corpus_{IT\_Train}$ and $Lexicon_{IT}$ *are* used to obtain the delimiter list of the IT domain, $DList_{IT}$. $Corpus_{Legal\_Train}$ and $Lexicon_{Legal}$ are used to obtain the delimiter list of the legal domain, $DList_{Legal}$.

3.4 Experiment Implementation

At first, two unsupervised algorithms are applied to generate seeds. In *TF-IDF*, a general-purpose Chinese segmenter (Zhang et al., 2003) is first used to segment the domain corpus. All the segmented n-grams (1≤n≤4) are taken as term candidates. The first seed set S_1 is collected which consists of 500 terms with the highest *TF-IDF* scores and 500 non-terms with the lowest *TF-IDF* scores. In TE_{Del}, a simple stop word list, $DList_{SW}$, without training is used to verify that the *FTL* approach works with no manually labeled data even in the seed generation step. $DList_{SW}$ takes 494 general purpose stop words downloaded from a Chinese NLP resource website (www.nlp.org.cn) without any modification. The second seed set S_2 is collected which consists of 500 terms and 500 non-terms the most confidently predicted by TE_{Del}.

Each seed set is verified using the classifier trained by the other seed set after generation. The classifier C_1 trained by S_1 is used to label S_2. Instances with different tags are removed from S_2. S_1 is similarly verified using C_2. The accuracies of S_1 and S_2 are further improved by verification.

As described before, *p* terms and *n* non-terms the most confidently predicted by the classifiers are considered correct and added to the seed set for the next iteration. A set of experiments are conducted to compare the performance of the proposed *FTL* approach by using different ranges of (*p*, *n*). The experimental results, which are not shown in this paper due to the space limitation, indicate that balanced growth of (*p*, *n*) is more efficient than unbalanced growth. We therefore set p=n=50 for the following experiments.

3.5 Evaluation on Term Extraction

An evaluation is conducted for term extraction on Chinese IT domain using $Corpus_{IT\_Test}$. Figure 3 summarizes the performance of the proposed *FTL* approach and the baseline methods. *FTL* achieves 81% precision when the number of extracted terms reaches 5,000. It is the best performance compared to the reference algorithms.

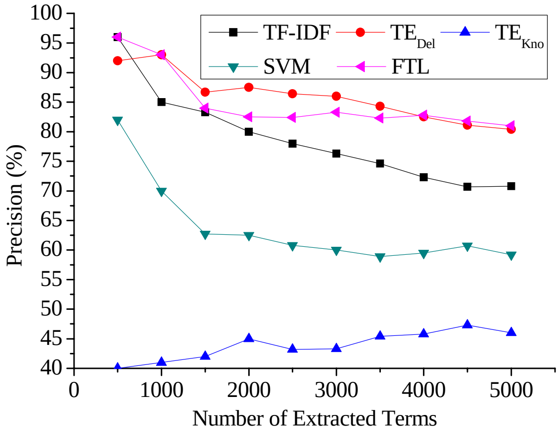

Figure 3: Performance on IT domain

The proposed *FTL* approach provides more than 20% higher performance compared to the *SVM* classifier which uses manually checked seeds. This indicates that the proposed model is benefit from prediction and verification based on two different classifiers. *FTL* requiring no explicit domain knowledge performs much better than TE_{Kno} relying on the 144K $Lexicon_{PKU}$ for term extraction. *FTL*, which applies TE_{Del} using a stop word list for seeds generation, also performs slightly better than TE_{Del} itself using delimiter list obtained from training data. The results reveal that *FTL* uses much less resources and still improve performance of term extraction.

Figure 4 shows the performance for the same set of algorithms using the legal corpus $Corpus_{Legal\_Test}$. The improvement in the legal domain shows a similar performance and trend. The proposed *FTL* performs the best. It achieves 72.4% precision when the number of extracted terms reaches 5,000. The performance is 3% to 13% higher in precision for the 5,000 extracted terms compared to the reference algorithms. This indicates that the proposed *FTL* approach is efficient for distinguishing terms from non-terms in different domains. Since a large lexicon in the Chinese legal domain is not available, the reference term verification algorithm TE_{Kno} does not even work. However, the proposed *FTL* approach using no prior domain knowledge still achieves a similar level of improvement. The results confirm that the proposed approach is quite stable across domains. In fact, the proposed approach can be easily applied to different domains since it requires no training data and no prior domain knowledge.

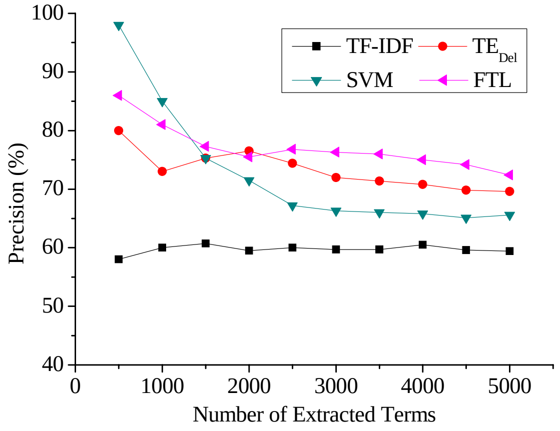

Figure 4: Performance on legal domain

There are three main reasons for the performance improvements of the proposed *FTL* approach. Firstly, the quality of automatically generated seeds is relatively good since *TF-IDF* and TE_{Del} perform well when the numbers of extracted terms are small. Moreover, verification is applied from the beginning which further improves the accuracy of the generated seeds. The average accuracy of the verified seeds is 92.9% which makes them useful for training. Secondly, the applied classifier can cover more kinds of terms since different types of features are integrated in each classifier. Thirdly, prediction and verification based on two classifiers guarantee the reliability of the extracted terms. Two relatively independent views are obtained by using different sets of seeds which makes the double check process more efficient. The fact that the *FTL* approach performs better over *SVM* using manually checked seeds further proves it.

4 Conclusion

In conclusion, this paper presents a *Fault-Tolerant Learning* approach for term extraction. The main purpose of *FTL* is to train classifiers for prediction and verification using noisy seeds. A fully automatic learning process is constructed by using automatically labeled instances. The proposed approach has some theoretical advantages. *FTL* integrates different types of features to cover as many kinds of terms as possible. It separately trains two classifiers for double check in order to filter out noise. Moreover, *FTL* requires no prior domain knowledge and no training data. Thus it can be applied to different domains much more easily than traditional supervised methods or domain knowledge based algorithms.

Experiments for term extraction are conducted on IT domain and legal domain, respectively. Evaluations indicate that the proposed approach can improve precision of term extraction quite significantly. The fact that the proposed approach achieves the best performance on two different domains verifies its domain independent nature.

The motivation of *FTL* is that learning can be done without manually checked seeds. *FTL* achieves the best performance on Chinese term extraction which indicates the efficiency of the proposed approach. Furthermore, the proposed *FTL* approach can also be applied to other resource-limited tasks if the assumptions described in section 2.1 can be satisfied.

References

Ando Rie K. and Tong Zhang. 2005. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *The Journal of Machine Learning Research*, 6, 1817-1853.

Blum A. and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp.92-100.

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

Chen, Yirong, Qin Lu, Wenjie Li, Zhifang Sui and Luning Ji. 2006. A Study on Term Extraction Based on Classified Corpora. *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Eibe Frank, Gordon. W. Paynter, Ian H. Witten, Carl Gutwin and Craig G. 1999. Nevill-Manning. Domain-specific keyphrase Extraction. *Proceedings of 16th International Joint Conference on Artificial Intelligence*, pp.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.

Hisamitsu T. and Y. Niwa. 2002. A measure of term representativeness based on the number of co-occurring salient words. *Proceedings of the 19th International Conference on Computational Linguistics*.

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

Ji Luning and Qin Lu. 2007. Chinese Term Extraction Using Window-Based Contextual Information. *Proceedings of Conference on Intelligent Text Processing and Computational Linguistics 2007, LNCS 4394*, pp.62-74.

Li Hongqiao, Chang-Ning Huang, Jianfeng Gao, and Xiaozhong Fan. 2004. The use of SVM for Chinese new word identification. *Proceedings of the First International Joint Conference on Natural Language Processing*, pp.723-732.

Laprie J.C.. 1985. Dependable Computing and Fault Tolerance: Concepts and Terminology. *Proceedings of the 15th IEEE Intternational Symposium on Fault-Tolerant Computing*, pp.2-11.

Luo Shengfen, and Maosong Sun. 2003. Two-Character Chinese Word Extraction Based on Hybrid of Internal and Contextual Measures. *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pp.24-30.

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

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? *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing*.

Sornlertlamvanich V., Potipiti T., and Charoenporn T. 2000. Automatic corpus-based Thai word extraction with the C4.5 learning algorithm. *Proceedings of Proceedings of the 18th International Conference on Computational Linguistics*.

Vapnik VN. 1995. The Nature of Statistical Learning Theory. Springer, 1995.

Yang Yuhang, Qin Lu and Tiejun Zhao. 2008. Chinese Term Extraction Using Minimal Resources. *Proceedings of the 22th International Conference on Computational Linguistics,* pp.1033-1040.

Yang Yuhang, Tiejun Zhao, Qin Lu, Dequan Zheng and Hao Yu. 2009. Chinese Term Extraction Using Different Types of Relevance. *Proceedings of* joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp.213-216.

Zhang Hua-Ping, Hong-Kui Yu, De-Yi Xiong and Qun Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. *Proceedings of the second SIGHAN workshop on Chinese language processing,* pp.184-187.

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