Iterative Chinese Bi-gram Term Extraction Using Machine-learning Classification Approach

Chia-Ming LEE[1], *Chien-Kang HUANG*[1], *Kuo-Ming TANG*[1]
(1) Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei, Taiwan (R.O.C.)
trueming@gmail.com, ckhuang@ntu.edu.tw, d965251013@ntu.edu.tw

ABSTRACT

This paper presents an iterative approach to extracting Chinese terms. Unlike the traditional approach to extracting Chinese terms, which requires the assistance of a dictionary, the proposed approach exploits the Support Vector Machine classifier which learns the extraction rules from the occurrences of a single popular term in the corpus. Additionally, we have designed a very effective feature set and a systematic approach for selecting the positive and negative samples as the source of training. An ancient Chinese corpus, Chinese Buddhist Texts, was taken as the experiment corpus. According to our experiment results, the proposed approach can achieve a very competitive result in comparison with the Chinese Knowledge and Information Processing (CKIP) system from Academia Sinica.

KEYWORDS : Chinese Term Extraction, Iterative Term Extraction, Support Vector Machine (SVM), Contextual Information, Single String Term Extraction.

1 Introduction

Chinese Term Extraction (CTE) is a fundamental issue in Chinese natural language processing studies. Existing research on term extraction evolved from two different types of corpora: pure-text corpora and labeled corpora (L.-F. Chien, 1997). In addition to the pure text, the labeled corpora contain an abundance of known information, such as terms, parts of speech, or even the syntactic tree structure. Grammar-based term extraction approaches are conventionally used with a labeled corpus (L.-F. Chien, 1997), and character-statistics-based term extraction approaches are usually used with pure-text corpora. However, both existing term extraction approaches need additional known information, prepared as supplements, which are independent from the corpus itself, and on principle, the more known information is prepared, the better the quality of the extracted terms. For instance, terms and part-of-speech labels within labeled corpora and dictionaries for verifying candidate terms found during the process of extracting terms from a pure-text corpus are usually additional prepared information.

Because all existing studies require more known information, this paper proposes the design of a process that extracts terms from a pure-text corpus without any additional information, especially known terms. In order to extract terms without a dictionary or other supplemental information, we introduce an iterative machine-learning term-extraction process. In this process, a Support Vector Machine (SVM) is exploited as the core of our learning mechanism. In order to extract terms starting from one specified term, the first step of our proposed approach is to generate the positive and negative examples with the one specified term and train an optimized SVM classification model with the proposed “term features” from an experimental corpus. Then, we can use the model to extract terms from the same corpus. In the iterative process, the previous results of the SVM term extraction are prepared as the training samples for the next SVM term extraction iteration. In our experiment, we specified one popular term as the initial learning sample and revealed the increasing rate of extracted terms of each iteration. The performance evaluation of the proposed approach was achieved by comparing the terms extracted using the iterative process with the terms verified by experts in chosen paragraphs from the experimental corpus.

In previous works, machine-learning algorithms have been used in many term-extraction studies. However, they have all extracted from labeled corpora with abundant known information (details are in the “Related Works” section). On the other hand, the concept of using an iterative process also has been proposed in other term-extraction research. For instance, the iterative model has been adopted by researchers studying the Sinica Corpus(Institute of Information Science and CKIP group in Academia Sinica, Since 1990). They used a segmentation tool to increase the term database, and then they further enhanced the segmentation tool with a larger term database. In our iterative approach, we have addressed a very extreme situation: starting from a single specified term.

In order to focus on the proposed idea of generating a learning sample, adding a contextual information extension, and developing an iterative extraction process, we executed the bi-gram CTE on a Middle Ages Chinese corpus, because bi-gram terms are the major part of Chinese texts (A. Chen, He, Xu, Gey, & Meggs, 1997) and also of the corpus used in our experiment[1].

[1] The book index of the corpus we used for our experiment [15] listed 404 uni-gram terms, 2678 bi-gram terms, 2227 tri-gram terms, 2404 quadri-gram terms, and 3334 other terms.

From the perspective of the machine-learning approach, terms with different lengths in a CTE might have some unique features that provide better classification results, and we leave studying these to our future work.

2 Related Works

Existing CTE research developed from studies using two types of corpora: labeled corpora and pure-text corpora. Labeled corpora are word-segmented corpora and usually contain abundant grammar information, such as parts of speech and even syntactic structures. The major purpose of a labeled-corpus CTE is to identify unknown words, also called out-of-vocabulary (OOV) terms, and to solve word-segmentation mistakes caused by natural language ambiguity problems in a corpus. Because labeled corpora contain grammar information, labeled-corpus CTE studies usually applied the grammar-based approaches. Other studies conducted serial unknown word identification on the Sinica Corpus (Sinica, Since 1990) using morphology rules (K.-J. Chen & Bai, 1998; K.-J. Chen & Ma, 2002; Ma & Chen, 2003).

The other type of corpora is pure-text corpora, which are growing with increased usage of text digitization and the World-Wide Web. The purpose of conducting a CTE on pure-text corpora is to recognize keywords, key phrases (L.-F. Chien, 1997; L. Chien, 1999), or domain-specific terms in a specific corpus. Because pure-text corpora are not accompanied by additional known information, character statistic-based approaches were usually used for CTEs of pure-text corpora. Therefore, many statistical patterns of Chinese terms have been proposed. For example, Chien's "Completed Lexicon Pattern" (L.-F. Chien, 1997) is a simple and effective design of Chinese terms in context, in which association and left and right context dependency were included.

2.1 Existing Machine-Learning Algorithms Used for Term Extraction

Many machine-learning algorithms are used in the study of CTE. The Hidden Markov Model (HMM) was used widely, because it is effective for modeling natural languages. Yu et al. (Yu, Zhang, Liu, Lv, & Shi, 2006) used different layers of HMMs to identify names of people, locations, and organizations in the ICTCLAS (Institute of Computing Technology Chinese Academy of Sciences, Since 2002) corpus. Cen et al. trained dual HMMs—a POS labeling model and a term boundary labeling model—to extract domain-specific terms (Cen, Han, & Ji, 2008). Xie et al. used a lexical chain of semantic relationships to extract key phrases from news-oriented Web pages (Xie, Wu, Hu, & Wang, 2008). However, all of the existing CTE research applied HMMs on labeled corpora, because labeled corpora have additional information, such as parts of speech or semantic relationships, which correspond to node-state patterns in the HMM.

Another popular machine learning algorithm used for CTE is the Support Vector Machine (SVM). Different from an HMM, the SVM is a multi-vector classification algorithm (Boser, Guyon, & Vapnik, 1992), and it is also a 2-phrase algorithm that employs a model-training phrase and a model-using (predicting) phrase. The major task of using the SVM is selecting a learning sample and a sample feature. Several researchers have used the SVM for CTE with the CoNLL-2000 Chunking Corpus (Sang & Buchholz, 2000): Goh used the SVM for recognizing person's names (Ling, Asahara, & Matsumoto, 2003). It used family names, contextual chunked marks and parts of speech as sample features. Li et al. extracted bi-gram and tri-gram terms using the SVM (Li, Huang, Gao, & Fan, 2005). The sample features they used were in-word probability of a character (IWP), analogy to new words, anti-word list, and frequency. The previously mentioned

CTE research studies still conducted extraction with a labeled corpus. However, this paper proposes a process to extract terms in a pure-text corpus using the SVM, and it also proposes a method of selecting a learning sample and a feature without additional known information.

3 Iterative Machine-Learning Term Extraction

Under the precondition of performing extraction without known information other than texts, the first problem in this machine-learning extraction process is how to generate sufficient number of good representational learning samples and sample features from a closed corpus. This section introduces a "negative sample selection" process, which fulfills the need to have a high capacity of differentiation in the known information. Also, a sufficient number of contextual lexicon parameters were added into the sample features in order to heighten the differentiation ability of the known information.

When considering the iterative process, we designed the process to increase the number of learning samples by filtering the extraction results during each iteration of the process. Therefore, this section will discuss three iterative issues: initial learning sample selection, iterative learning sample filtering, and the last convergence conditions of the entire iterative process.

3.1 Structure of the Iterative Machine-Learning Term-Extraction Process

There are six steps within the three phases of the iterative machine-learning term extraction process (see numbers 1 through 6 in the chart below). The first phase, also Step 1, is the initial selection of known information, the one known term. The second phase, the SVM machine-learning term extraction, includes Steps 2, 3, and 4. In Step 2, the SVM input dataset, in which data is in the form of positive/negative sample-features, is transformed and generated from the known information. The output of Step 3, SVM model training, is a sample classification model, which is used in step 4, SVM predicting, to extract terms from unknown strings in the corpus. Unknown strings, in the bi-gram extraction process, are all bi-gram strings in the corpus, and all unknown strings also need to be translated into a sample-features dataset for SVM predicting.

The iterative process stops when the increase of the term sizes between two subsequent SVM outputs is less than 2%, and it is checked in Step 5. For the next round of the iterative extraction process, Step 6 filters out initial known terms from the SVM output terms. The third phrase, evaluation, starts when the iterative process stops. The evaluation method compares the extracted terms from the iterative process with verified terms by experts and judges their precision, recall rate, and f-measure. The iterative processing chart is depicted in Figure 1.

3.2 Initial Known Information Selection

In Step 1, initial selection, the most frequent term in the corpus used in the experiment was selected as the initial known term. Although there was only one initial term, the available number of actual learning samples is the same as the word frequency of the initial term, because learning samples are contextually dependent. In this way, we were able to ensure the appropriate size of the initial learning samples.

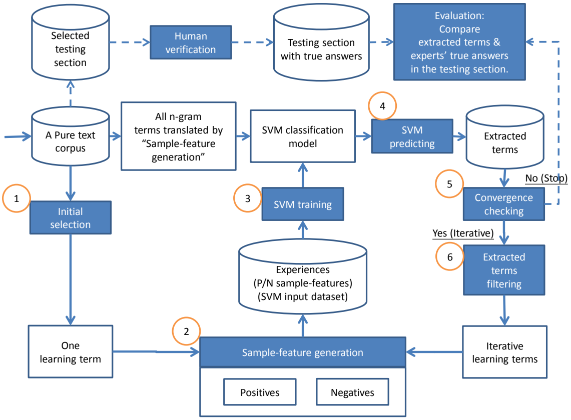
Selected testing section
Human verification
Testing section with true answers
Evaluation: Compare extracted terms & experts' true answers in the testing section.
A Pure text corpus
All n-gram terms translated by "Sample-feature generation"
SVM classification model
4
SVM predicting
Extracted terms
No (Stop)
5
Convergence checking
1
Initial selection
3
SVM training
Yes (Iterative)
6
Extracted terms filtering
Experiences (P/N sample-features) (SVM input dataset)
2
One learning term
Sample-feature generation
Positives
Negatives
Iterative learning terms

FIGURE 1 – The iterative machine-learning term-extraction process

3.3 Sample-feature Generation

Sample-feature Generation, step 2, generates learning samples from the initial known term(s) and feature parameters from each learning sample. There are three main components of sample-feature generation: (1) positive and negative learning sample generation, (2) lexicon features selection, and (3) contextual information extension.

3.3.1 Positive and Negative Sample Selection.

In this term extraction process, learning samples include positive samples and negative samples. Positive samples are the strings in which selected initial term(s) are located in a corpus. Negative samples are the new strings produced from shifting one character to the right or to the left in the context of the positive samples. In this way, every positive sample typically comes with two negative samples, and this can increase the number of learning samples, which is beneficial for the processing of the SVM classification algorithm. More importantly, shifting one character in context can break the lexicon pattern of the original sample, and this fulfills the need to have a high capacity of differentiation within the learning samples. Figure 2 shows a positive sample and its two negative sample selections.

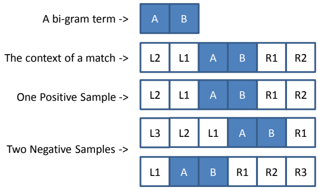

FIGURE 2 – Positive and negative sample selection

3.3.2 Features Selection

There are 10 types of features chosen for a learning sample in this extraction process, including frequency, the number of distinct characters to the left and right of the learning sample, the number of breaking symbols (non-Chinese characters and paragraph marks) to the left and right of the learning sample, association, and the usage of freedom to the left and right of the learning sample. Among these features, association and the usage of freedom (also called left and right context dependency) refer to the "estimation of complete lexical patterns" proposed by Chien (L.-F. Chien, 1997), as shown in Figure 3.

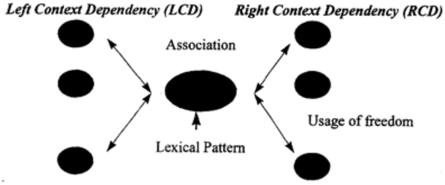

FIGURE 3 – The estimation of complete lexical patterns (L.-F. Chien, 1997)

Each of the lexical patterns is estimated using Equations 1, 2, and 3, as described below:

$$\text{Association (AEc)} = f(x) / (f(y) + f(z) - f(x)) \quad (1)$$

Where x is the lexical pattern to be estimated; $x = x_1, x_2, \ldots, x_n$, y and z are the two longest composed substrings of x with length $n-1$; $y = x_1, \ldots, x_{n-1}$, $z = x_2, \ldots, x_n$. Then, $f(x)$ is the frequency of x, $f(y)$ is the frequency of y and $f(z)$ is the frequecny of z.

$$\text{Left Context Dependency (LCD)} = f(\max\_x_L) / f(x) \quad (2)$$

$f(\max\_x_L)$ is the maximum frequency of the occurrence of distinct characters to the left of the lexical pattern x.

$$\text{Right Context Dependency (RCD)} = f(\max\_x_R) / f(x) \quad (3)$$

$f(\max\_x_R)$ is the maximum frequency of the occurrence of distinct characters to the right of the lexical pattern x.

The list of the 10 types of features are shown in Table 1.

1.	Frequency
2.	The number of distinct character to the left
3.	The maximum occurrence frequency of distinct character to the left
4.	The number of breaking symbols to the left
5.	The number of distinct character to the right
6.	The maximum occurrence frequency of distinct character to the right
7.	The number of breaking symbols to the right
8.	Association: AEc
9.	Left-context dependency (LCD)
10.	Right-context dependency (RCD)

TABLE 1 – Feature list

3.3.3 Contextual Feature Extension

The concept of contextual information has often been used in information extraction research as well as in existing CTE research for entity identification (Ji, Sum, Lu, Li, & Chen, 2007; Ling et al., 2003). In order to enhance the effect of contextual information on the classification results, in our study, the initial learning samples (also called original samples) were designed to extend into longer learning samples. Then, the features of every bi-gram within the new samples, as shown in Table 1, are collected as the contextual information of the original samples. According to the result of the experiments, not included in this paper, the best length of the longer learning samples was two characters on both the left and right sides. For instance, an original bi-gram learning sample will extend to become a 6-gram learning sample, and the feature parameter will become five times larger than before, because there will be 5 bi-grams within the new sample. Table 2 shows the differences of sample lengths and feature parameters before and after the contextual extension.

Before	After
Original n-gram learning sample: A B	Extended learning sample: L2 L1 A B R1 R2
Contextual bi-gram set: 0	Contextual bi-gram set: 4 L2 L1 L1 A B R1 R1 R2
Number of original features: 10 10 features (see Table 1) for A B.	Number of original features: 10 10 features (see Table 1) for A B. extend Number of contextual info. features: 40 10 features (see Table 1) for each extended bi-gram, L2 L1 L1 A B R1 R1 R2

TABLE 2 – Contextual feature extension

In addition to the features in Table 2, to increase the number of features of the contextual information, we also added into the feature set all uni-gram frequencies within an extended learning sample. Therefore, an extended bi-gram learning sample, a 6-gram string, will actually have a total of 56 features, including six uni-gram frequencies and 50 features coming from five bi-gram feature sets.

3.4 SVM Machine-Learning Term Extraction

The Support Vector Machine (SVM) algorithm constructs a hyper-plane in a high-dimensional space for classification or other tasks. A good separation is achieved by the hyper-plane farthest from the nearest training data point of any class.

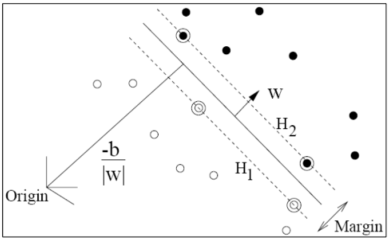

FIGURE 4 – Support Vector Machine (SVM)

In Figure 4, W is the good separation (the classification hyper-plane) of the two classes—white spots and black spots—and H1 and H2 are support hyper-planes.

$$\mathbf{W}^T\mathbf{X}+b=0, \qquad \text{H1: } \mathbf{W}^T\mathbf{X}+b=1, \qquad \text{H2: } \mathbf{W}^T\mathbf{X}+b=-1$$

To maximize the distance between H1 and H2 (2 / ||w||):

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i\left[y_i\left(w^T x_i+b\right)-1\right]$$

In our study, libsvm tools (Chang & Lin, 2011) were used to execute the SVM algorithm during the term-extraction process. The SVM algorithm includes two phases: model training and unknown-term predicting. In the model-training phase, the input data is the sample-feature dataset of learning samples generated from the sample feature generation step, and the output data is a classification model file. Meanwhile, all strings (not distinct) with the same length as the learning samples in the corpus are considered unknown samples. Unknown samples will convert to the sample-feature dataset in the exactly same way learning samples generation converts to its dataset. In the unknown-term-predicting phase, the input data is the sample-feature dataset of unknown samples and the classification model file output from the earlier phase, and the output data are unknown strings that are predicted as terms. The predicted output data are the term extraction results.

3.5 Iterative Convergence Condition

The stop condition of the iterative extraction process occurs when the increase between the term sizes from two subsequent extraction results is less than 2%.

3.6 Iterative Learning Sample Selection

Step 6, filtering, in Figure 1, filters the extracted terms, which will become learning samples for the next extraction round. The libsvm tools provide the function of probability estimation, predicting the probability of each unknown sample being classified into a certain category. In this step, the rate of probability is then adjusted to filter out the learning samples for the next round. According to the default setting in libsvm, an unknown sample is considered a term when its predicted probability rate is greater than or equal to 50%. Also, in this iterative extraction process, the learning samples for the next round are chosen when the predicted probability rate is greater than or equal to 90%.

3.7 Evaluation of the Term Extraction Results

In this paper, precision, recall, and f-measure were used to evaluate the effectiveness of the term extraction results. The correct answers were pre-decided by experts for a chosen paragraph in the corpus and then compared with the extraction results. The experts did not examine the extracted results directly. The evaluation results are shown in the next section.

4 The Experiment

As part of a Chinese text archive from the Middle Ages provided by the Chinese Buddhist Electronic Text Association (CBETA), the collection of Saddharma Puṇḍarīka (Lotus of the true Dharma) was used as the experimental corpus, which consists of 16 sutras labeled from T0262 to T0277 in the Taisho Revised Tripitaka. This corpus contains 514,722 Chinese characters, among which are 3,041 distinct uni-grams, 82,688 distinct bi-grams, and 202,187 distinct tri-grams. Generally, ancient Chinese corpora are rarely used in term extraction research due to the difficulty of collecting known information. However, for our study, the design of the initial known information selection and the contextual sample-features generation in the iterative learning process produced good extraction results regarding the ancient Chinese corpus.

Another reason for using Buddhist texts is that the Middle Age Chinese literature, of which the Buddhist canon is representative, is mainly composed by bi-gram terms, the same as is current Chinese literature (梁曉虹, 2005). So, theoretically this iterative term extraction process can be directly applied to other forms of current Chinese texts as well. The experiment in this section focuses on bi-gram term extraction from the collection of Saddharma Puṇḍarīka (CBETA) to list the iterative experiment results and further compares them with the results of other term extraction algorithms.

In the corpus, Sàtánfēntuólìjīng, a sutra (T0265) from the collection of Saddharma Puṇḍarīka, was chosen to be the evaluation text. In Sàtánfēntuólìjīng, there are 1,430 non-distinct bi-grams which have been prepared with experts' judgments of true bi-gram terms. The ratio size of the evaluation text is 0.32% of the entire corpus, which consists of 442,513 non-distinct bi-grams.

4.1 Iterative Term Extraction

This section shows the experimental results of the iterative extraction process. The size of the experimental corpus is 2,058,888 bytes. There are a total of 514722 symbols encoded by UTF-32, and 442513 non-distinct Chinese bi-grams in the corpus. The initial selected bi-gram term is 一切 and the number of initial learning samples are 2387, with the term frequency of 一切 in the corpus.

Table 3 shows the results of numbers of initial learning samples and SVM extraction terms, and increasing ratios of total and unique extraction terms in 8 rounds.

In Table 3, values in the "Extraction ratio" row are determined by the following equation:

$$\frac{\text{number of "SVM extraction terms"}}{442513\left(\text{the number of total bi-grams in the corpus used for our experiment}\right)}$$

Iterative Rounds	1	2	3	4	5	6	7	8
Initial learning samples	2387	18259	55360	120826	148781	160849	166583	169777
SVM extraction terms	18259	55360	120826	148781	160849	166583	169777	171570
Extraction ratio	4.126%	12.51%	27.30%	33.62%	36.34%	37.64%	38.36%	38.77%
Subsequent increasing ratio		+8.39%	+14.79%	+6.32%	+2.72%	+1.30% (<2.00%)	+0.72% (<2.00%)	+0.41% (<2.00%)
Unique extraction terms	15	1044	8148	12287	14679	16422	17520	18174
Unique terms increasing times		*69.60	*7.80	*1.50	*1.19	*1.12	*1.06	*1.04

TABLE 3 – The extraction convergence table

The "Subsequent increasing ratio" row shows increasing ratios of the "Extraction ratio" row. The "Unique extraction terms" row indicates the distinct number of "SVM extraction terms," and the "Unique terms increasing times" row shows the number of times the number of extracted distinct terms increased when compared to the previous round.

Based on the iterative experiment results, the sixth round should be the final extraction result in this experiment, because the saturation condition is set when the increase in the number of terms from one SVM to the next is less than 2%. Meanwhile, the states of the increase in number of the SVM extraction terms, the total bi-gram terms, and the Unique extraction terms are different. The increase of unique terms is about to saturate in the third round, which is two or three rounds before the saturation of the increase of total extraction terms.

4.2 Evaluation

In the corpus, Sàtánfēntuólìjīng (Tripitaka sutra number T0265) was chosen to be the texting data. There are 1,430 non-distinct Chinese bi-grams in Sàtánfēntuólìjīng, and the ratio size of the texting data is 0.32% of the entire corpus. In the testing data, the number of positive samples (expert verified bi-gram terms) is 332, and the number of negative samples (expert verified non-bi-gram terms) is 1098. The evaluation indices are listed below.

$$Precision = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive}$$

$$Recall\ = \frac{True\ Positive}{True\ Positive\ +\ False\ Negative}$$

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Iterative Rounds	True Positive	False Positive	False Negative	True Negative
1	26	3	306	1095
2	85	24	247	1074
3	201	128	131	970
4	235	188	97	910
5	251	219	81	879
6	255	228	77	870
7	256	234	76	864
8	257	238	75	860

TABLE 4 – Table of the classifier prediction compared to the experts' judgments

Iterative Rounds	Accuracy	Precision	Recall	F-measure
1	78.4%	89.7%	7.8%	14.35%
2	81.0%	78.0%	25.6%	38.54%
3	81.9%	61.1%	60.5%	60.79%
4	80.1%	55.6%	70.8%	62.28%
5	79.0%	53.4%	75.6%	62.58%
6	78.7%	52.8%	76.8%	62.57%
7	78.3%	52.2%	77.1%	62.25%
8	78.1%	51.9%	77.4%	62.13%

TABLE 5 – The evaluation calculation

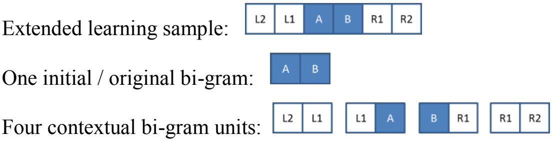
Extended learning sample:
L2 L1 A B R1 R2
One initial / original bi-gram:
A B
Four contextual bi-gram units:
L2 L1
L1 A
B R1
R1 R2

FIGURE 5 – The contextual bi-gram set

No.	Feature Explanation	F-source
1	RCD of A B	0.4420
2	LCD of A B	0.3304
3	LCD of B R1	0.3281
4	RCD of L1 A	0.3144
5	The number of distinct character to the left of A B	0.2284
6	The number of distinct character to the right of A B	0.2199
7	AEc of A B	0.2108
8	The number of distinct character to the left of B R1	0.1598
9	RCD of B R1	0.1513
10	The number of breaking symbols to the left of A B	0.1480
11	LCD of L1 A	0.1390
12	The number of distinct character to the right of L1 A	0.1338
13	The number of distinct character to the right of B R1	0.1295
14	Frequency of A B	0.1256
15	The number of distinct character to the left of L1 A	0.1123

TABLE 6 – Top 15 features sorted by f-score

4.3 Features Selection Analysis

This section analyzes the importance of features used in the SVM extraction process. A total of 56 features were calculated and sorted by the f-score algorithm proposed by Chen and Lin's SVM feature-selected research (Y.-W. Chen & Lin, 2006).

Figure 5 shows that one extended learning sample is represented by 5 bi-grams, including one original bi-gram and 4 contextual bi-grams. And Table 6, the top 15 features of training dataset, shows that the learning sample and contextual features extension has a great influence on the SVM classification extraction.

5 Comparison

This section discusses the comparison of the bi-gram term extraction results from three different term extraction algorithms with the verification answers by experts using the same corpus. The first set of results come from the iterative machine learning extraction process proposed by this paper. The second set is the control group using the same process with two controlled conditions: (1) using 2,678 known terms from the book index of the corpus (大藏經學術用語研究會, 198-?) as the initial known terms and (2) using no iterative extraction and executing the SVM extraction process only once. With this control group, we can compare the extraction differences between using known terms outside of the corpus and known terms trained iteratively inside of the corpus. The third extraction algorithm used for comparison is the Sinica CKIP system (Chinese Knowledge Information Processing Group), which represents the term segmentation and extraction algorithms based on a "term database."

In Table 7, the "MLTE iterative" row using the sixth round extraction results from the previous experiment, and the "MLTE_dic" row is the extraction result of the dictionary control group. The comparison table shows that the SVM algorithm with dictionary-known terms more effectively extracted terms, but the iterative SVM algorithm extraction had the highest recall rate, and both extraction results from the SVM algorithm had a higher F-measure values than the CKIP recognition results on bi-gram.

	Precision	Recall	F-measure
MLTE_dic	65.8%(202/307)	60.8%(202/332)	63.20 %
MLTE_iterative	52.8%(255/483)	76.8%(255/332)	62.58 %
CKIP	62.5%(203/325)	61.1%(203/332)	61.80 %

TABLE 7 – Comparison of 3 extraction results

6 Discussions

For this study, we proposed an iterative machine-learning term-extraction process that does not use known information other than the pure-text corpus itself. In this process, the effective selection of learning samples and sample features were designed specifically for two classes of SVM machine-learning algorithms. Based on the experimental results of this iterative extraction process, there are some issues and problems deserving further discussions:

1. The design of the process to extract n-gram terms of any given length. The current iterative process in this paper can only extract fixed-length n-gram terms. In order to extract terms of

any length and integrate the extraction results of terms of varied lengths, it is more ideal to directly design a method to extract samples and sample feature parameters for terms of any given length and to establish the extraction model.

2. The testing and discussion of the applicability of this iterative extraction process with different types of corpora, such as current Chinese texts or Web messages.
3. The discussion of the initial learning sample selection, which is a changeable parameter in the initial learning sample in this extraction process. Besides the most frequent terms in the corpus, if we use other high frequency terms or if we take into account other factors, such as parts of speech the process might provide different extraction results and issues for further discussion.

Acknowledgments

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC 99-2410-H-002-203-MY2.

References

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). *A training algorithm for optimal margin classifiers*. Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, United States.

Cen, Y., Han, Z., & Ji, P. (2008). *Chinese Term Recognition and Extraction Based on Hidden Markov Model.* Paper presented at the Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol., 2*(3), 1-27. doi: 10.1145/1961189.1961199

Chen, A., He, J., Xu, L., Gey, F. C., & Meggs, J. (1997). *Chinese text retrieval without using a dictionary*. Paper presented at the Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, United States.

Chen, K.-J., & Bai, M.-H. (1998). *Unknown Word Detection for Chinese by a Corpus-based Learning Method*. Paper presented at the Computational Linguistics and Chinese Language Processing.

Chen, K.-J., & Ma, W.-Y. (2002). *Unknown word extraction for Chinese documents*. Paper presented at the Proceedings of the 19th international conference on Computational linguistics - Volume 1, Taipei, Taiwan.

Chen, Y.-W., & Lin, C.-J. (2006). Combining SVMs with Various Feature Selection Strategies. *Feature Extraction - Studies in Fuzziness and Soft Computing, 207*, 315-324.

Chien, L.-F. (1997). *PAT-tree-based keyword extraction for Chinese information retrieval*. Paper presented at the Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, United States.

Chien, L. (1999). PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. *Information Processing and Management, 35*(4), 501.

Chinese Knowledge Information Processing Group. CKIP Chinese Word Segmentation System, from http://ckipsvr.iis.sinica.edu.tw/

Institute of Computing Technology Chinese Academy of Sciences. (Since 2002). Institute of Computing Technology Chinese Lexical Analysis System (ICTCLAS). *ICTCLAS 2010*, from http://ictclas.org/

Institute of Information Science and CKIP group in Academia Sinica. (Since 1990). Academia Sinica Balanced Corpus of Modern Chinese. *Sinica Corpus 4.0*, from http://db1x.sinica.edu.tw/kiwi/mkiwi/

Ji, L., Sum, M., Lu, Q., Li, W., & Chen, Y. (2007). *Chinese Terminology Extraction Using Window-Based Contextual Information*. Paper presented at the CICLing 2007.

Li, H., Huang, C.-N., Gao, J., & Fan, X. (2005). The Use of SVM for Chinese New Word Identification

Natural Language Processing – IJCNLP 2004. In K.-Y. Su, J. i. Tsujii, J.-H. Lee & O. Kwong (Eds.), (Vol. 3248, pp. 723-732): Springer Berlin / Heidelberg.

Ling, G. C., Asahara, M., & Matsumoto, Y. (2003). *Chinese unknown word identification using character-based tagging and chunking*. Paper presented at the Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2, Sapporo, Japan.

Ma, W.-Y., & Chen, K.-J. (2003). *A bottom-up merging algorithm for Chinese unknown word extraction*. Paper presented at the Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17, Sapporo, Japan.

Sang, E. F. T. K., & Buchholz, S. (2000). *Introduction to the CoNLL-2000 shared task: chunking*. Paper presented at the Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7, Lisbon, Portugal.

Sinica, I. o. I. S. a. C. g. i. A. (Since 1990). Academia Sinica Balanced Corpus of Modern Chinese. *Sinica Corpus 4.0*, from http://db1x.sinica.edu.tw/kiwi/mkiwi/

Xie, F., Wu, X., Hu, X.-G., & Wang, F.-Y. (2008). *Keyphrase Extraction from Chinese News Web Pages Based on Semantic Relations*. Paper presented at the Proceedings of the IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics, Taipei, Taiwan.

Yu, H.-k., Zhang, H.-p., Liu, Q., Lv, X.-q., & Shi, S.-c. (2006). Chinese named entity identification using cascaded hidden Markov model. *Journal on Communications, 27-2*, 8.

大藏經學術用語研究會. (198-?). *大藏經索引*. 台北: 新文豐.

梁曉虹, 徐., 陳五雲. (2005). *佛經音義與漢語詞彙研究*. 北京: 北京商務印書館.