Automatic Construction of a Chinese Electronic Dictionary

[+]Jing-Shin Chang, [+]Yi-Chung Lin and [+*]Keh-Yih Su

{shin,lyc}@hermes.ee.nthu.edu.tw, kysu@bdc.com.tw

[+]Department of Electrical Engineering
Natural Language Processing Lab
National Tsing-Hua University
Hsinchu, Taiwan 30043, ROC

[*]Behavior Design Corporation
2F, No.5, Industrial East Road IV
Science-Based Industrial Park
Hsinchu, Taiwan 30077, ROC

ABSTRACT

In this paper, an unsupervised approach for constructing a large-scale Chinese electronic dictionary is surveyed. The main purpose is to enable cheap and quick acquisition of a large-scale dictionary from a large untagged text corpus with the aid of the information in a small tagged seed corpus. The basic model is based on a Viterbi reestimation technique. During the dictionary construction process, it tries to optimize the automatic segmentation and tagging process by repeatedly refining the set of parameters of the underlying language model. The refined parameters are then used to further get a better tagging result. In addition, a two-class classifier, which is capable of classifying an n-gram either as a word or a non-word, is used in combination with the Viterbi training module to improve the system performance.

Two different system configurations had been developed to construct the dictionary. The configurations include (1) a Viterbi word identification module followed by a Viterbi POS tagging module and (2) a two-class classification module as the postfilter for the above Viterbi word identification module.

With a seed of 1,000 sentences and an untagged corpus of 311,591 sentences, the performance for bigram word identification is 56.88% in precision and 77.37% in recall when the two-class classifier is applied to the word list suggested by the Viterbi word identification module. The Viterbi part of speech tag reestimation stage gives the figures of 71.16% and 71.81% weighted precision rates and 73.42% and 73.83% weighted recall rates for the 2 different configurations when using a seed corpus of 9676 sentences.

1. Introduction and System Overview

A large-scale electronic dictionary is the fundamental component to many natural language and spoken language processing applications such as spelling correction, grammar checking, text-speech conversion, intelligent Chinese input methods and machine translation. However, a large electronic dictionary for natural language processing may not be available. This is true for the current Chinese language processing community. One possible way is to convert a general dictionary into its electronic form. Firstly, however, a general dictionary may not be updated frequently to reflect the current status of language uses, and many new lexicon entries may not be available. Secondly, a general dictionary may be lack of certain field-specific terms such that the language processing system cannot make use of such unregistered words for a domain-specific application. Even with the term registered, it may not have the special syntactic or semantic annotations for a particular domain. For instance, a machine translation system for translating computer manuals may need to update its lexicon frequently to catch up with the constantly changing computer technologies. In this case, a general dictionary may not provide significant help in the particular domain.

Furthermore, the number of lexical entries in a practical electronic dictionary usually exceeds tens of thousands. Therefore, human involvement will be costly and time-consuming. Even though some supervised learning methods are possible for annotating text corpora (e.g., [Chang 93]), it is still a large burden for tagging a large training corpus, and such approaches usually assume the existence of a large dictionary. In addition, human involvement may introduce inconsistency in the lexicon entries. An automatic and unsupervised dictionary construction approach is thus highly desirable.

Because there are only a few manually constructed Chinese electronic dictionaries available for general domain [CKIP 90, BDC 93], the techniques for automatic construction of large-scale Chinese electronic dictionaries from text corpora will be exploited in this paper. One particular difference between Chinese text and English text is that there is no natural delimiters, like spaces, between Chinese words. A Chinese version of "This is a book.", for instance, will look like "Thisisabook." Therefore, extracting Chinese lexicon entries and annotating the lexicon entries are much more difficult than other languages. An automatic approach must be used first to segment the Chinese text corpus into segmented text for further processing.

In this paper, an automatic approach to constructing an electronic dictionary, which contains lexical entries and their possible parts of speech tags, is proposed. In particular, we will use a reestimation technique, a small tagged seed corpus and a large untagged corpus to construct the dictionary.

The word tokens embedded in a Chinese corpus can be acquired by segmenting the text corpus into word tokens with a reestimation technique. The reestimation technique, referred to as the Viterbi training procedure for words (VTW), is used mainly to find possible word n-grams by maximizing the likelihood of the segmentation patterns of the segmented text corpus. However, more information may be used to identify whether an n-gram is really a word entry. In this paper, we thus also propose a two-class classification (TCC) method for identifying the word entries; a character n-gram is classified as either a word or a non-word n-gram according to some useful features observed from the seed corpus and a large unsegmented corpus.

These two techniques, can be combined or used separately to form a system for automatic word identification. In one configuration, we use a Viterbi reestimation algorithm to find out a list of candidate words in the large, untagged text corpus. We then use discriminative features, which provide intra-word information and inter-character information for judging whether a candidate word is qualified as a true word.

The word segmentation patterns based on the dictionary extracted by the word reestimation process, or the two class classifier, or a concatenation of these two modules, are then automatically tagged with part of speech information with a part of speech reestimation method. The reestimation process for POS tagging will be referred to as a Viterbi Training process for Tags (VTT).

2. Automatic Construction of Electronic Dictionary with Reestimation Approach

The fundamental building blocks for the above-mentioned automatic Chinese electronic dictionary construction system contain the following modules: (i) automatic word extraction system, and (ii) automatic part-of-speech tagging system. Figure 1 shows the block diagram of such a system, where the word extraction system is shown to be a word segmentation module implemented with the Viterbi Training procedure for words.

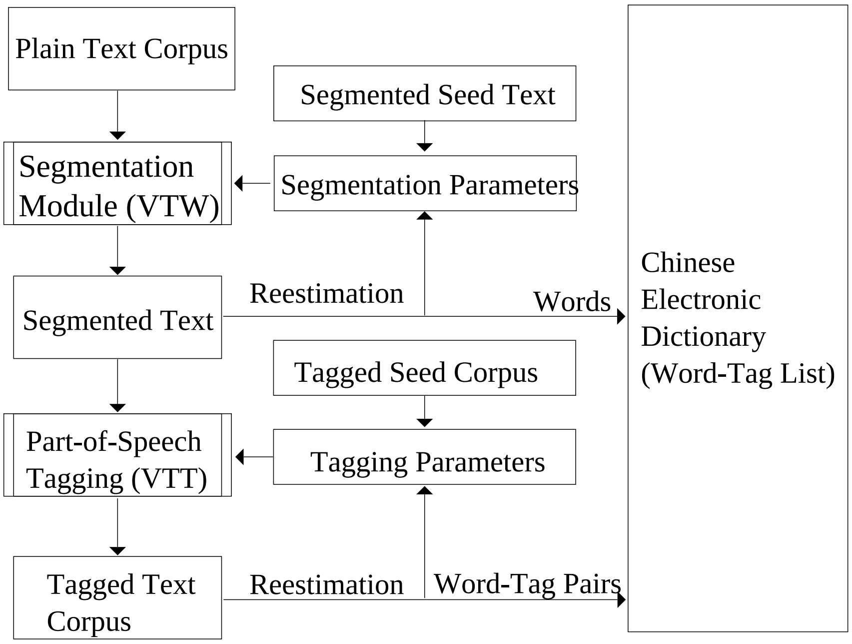

Figure 1 A Chinese Dictionary Construction System

The system reads a large untagged plain text and produces its segmented version based on a segmentation model (with or without TCC post-filtering). The main purpose of the segmentation module is to segment the Chinese text corpus into words because there is no natural delimiter between Chinese words in a text. After segmentation, each word in the segmented text is automatically tagged with its part of speech. The possible parts of speech for each word in the segmented plain text are then collected to form a POS annotated electronic dictionary.

A Viterbi reestimation process, as outlined below, could be used both for the word segmentation and POS tagging tasks to optimize the tagging patterns (including segmentation patters and POS tagging patterns) to a reasonable way. The principle is to find a set of initial segmentation or tagging parameters first from the small segmented or tagged seed corpus, and use this set of parameters to optimize the segmentation or POS tagging tasks. After the task is done, the *best* tagging pattern is updated, and the set of parameters are reestimated based on the distribution of the new tagging patterns and the seed. This process is repeated until a stopping criterion is met.

Since only the best tagging pattern for each sentence is used for reestimating the parameters, such a training procedure will be referred to as a Viterbi Training (VT) procedure, in contrast to an EM algorithm [Dempster 77], which considers all possible patterns and their expectations. Since an EM version of the training procedure may require a long computation time, we will leave this option to future research.

3. Automatic Word Identification: Viterbi Training for Words (VTW)

To compile an electronic dictionary (i.e., a word-tag list in the current task), we need to gather the word list within the corpus first. Since there is no natural delimiter, like space, between Chinese words, all the character n-grams in the text corpus are potential candidates for words. The first lexicon acquisition task is therefore to identify appropriate words embedded in the text corpus which are not known to the seed corpus. This task could be resolved by using a word segmentation model or a two-class classifier (to be described in the next sections).

Rule-based approaches [Ho 83, Chen 86, Yeh 91] as well as probabilistic approaches [Fan 88, Sproat 90, Chang 91, Chiang 92] to word segmentation had been proposed. For a large-scale system, the probabilistic approach is more practical when considering the capability of automatic training and cost. Practical probabilistic segmentation models can achieve quite satisfactory results [Chang 91, Chiang 92] provided that there is no unknown word to the system.

A particular segmentation pattern can be expressed in terms of the words they have. Given a string of Chinese characters $c_1, c_2, \ldots, c_n$, represented as c_1^n , a Bayesian decision rule requires that we find the best word segmentation pattern $\hat{W}$ among all possible segmentation patterns W_j which maximizes the following probability:

$$\hat{W} = \underset{W_j}{\operatorname{argmax}} P\left(W_j = w_{j,1}^{j,m_j} | c_1^n\right)$$

where $w_{j,1}^{j,m_j}$ are the m_j words in the j-th alternative segmentation pattern W_i. In the current task, we assume that there is only a small segmented seed corpus available. To reduce estimation error, we adopt the simple model used in [Chang 91]:

$$P\left(W_j = w_{j,1}^{j,m_j} | c_1^n\right) \cong \prod_{i=1}^{m_j} P\left(w_{j,i}\right)$$

which uses the product of word probabilities as the scoring function for segmentation. Other more complicated segmentation models [Chiang 92] may get better results. However, a more complicated model might not be appropriate in the current unsupervised mode of learning since the estimation error for the parameters may be high due to the small seed corpus. The following figure shows the block diagram of such a system.

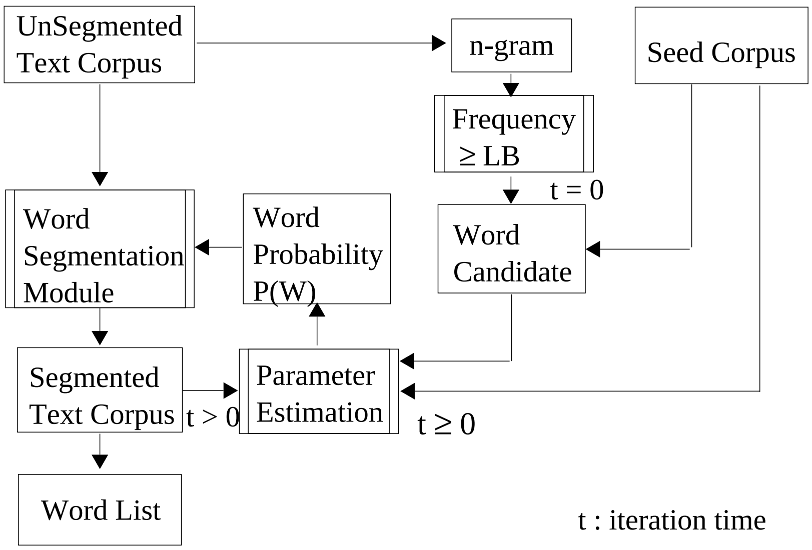

Figure 2 The block diagram of a Viterbi training model for word identification

Note the loop in re-estimating the word probabilities. Initially, the n-grams embedded in the unsegmented corpus is gathered to form a word candidate list. For practical purpose, we will only retain n-grams that are more frequent than a lower bound (LB=5), and only n-grams up to n=4 are considered (since most Chinese words are of length 1, 2, 3, or 4). The frequency lower bound restriction is applied to reduce the number of possible word candidates; it also removes n-grams that are not sufficiently useful even though they are judged as word candidates. Note that the words in the seed corpus are always included in the candidate list. In this sense, it plays the role of an initial dictionary. Furthermore, all the characters (1-grams) are included to avoid the generation of ‘unknown word regions’ in the segmented patterns.

Each word candidate will be associated with a non-zero word probability; the various segmentation patterns of the unsegmented corpus are then expanded in terms of such word candidates. The path (i.e., the segmentation pattern) with the highest score as evaluated according to the initial set of parameters (i.e., word probabilities) is then marked as the best path for the current iteration. A new set of parameters are then re-estimated based on the best path. This process repeats until the segmentation patterns no more change or a maximum number of iteration is reached. We then derive the word list to be included in the electronic dictionary from the segmented text corpus.

Initially, the word probability $P(w_{j,i})$ is estimated from the small tagged seed corpus. In the reestimation cycle, both the seed corpus and the segmented text corpus acquired in the previous iteration are *jointly* considered to get a better estimation for the word probabilities.

4. Automatic Word Identification: A Two-Class Classification (TCC) Model

The word list acquired through the above reestimation process is based on the optimization of the likelihood value of the word segmentation pattern in a sentence, which implicitly takes the contextual words into account. However, it may not take into account the features for forming a word from characters. It is desirable, for instance, to take some "strength" measures for the chunks of characters into account in order to know whether an n-gram is a word. Therefore, an alternative approach, which could also be used to supplement the VTW reestimation approach, is a Two-Class Classification model for classifying the character n-grams into words and non-words.

To identify whether an n-gram belongs to the word class (w) or the non-word class ($\overline{\mathrm{w}}$), each n-gram could be associated with a feature vector $\vec{x}$ observed from the large untagged corpus. It is then judged to see whether it is more likely to be generated from a word model or a non-word model based on $\vec{x}$.

To simplify the the design of the classifier, we use a simple linear discrimination function for classification:

$$g(\vec{x}_s, \vec{w}_s) = \vec{w}_s \bullet \vec{x}_s$$

where $\vec{x}_s$ is the feature vector (or score vector) and $\vec{w}_s$ is a set of weights, acquired from the seed corpus, for the various components of the score vector. An n-gram will be classified as a word if the weighting sum of $\vec{w}_s$ and $\vec{x}_s$ is greater than zero (or larger than a threshold λ_0). (For better results, a score vector derived from a log-likelihood ratio test as in [Su 94] could be used. Such an approach is being studied.)

For estimating the weights, the seed n-grams are firstly separated into the word and non-word classes by checking them against the known segmentation boundaries in the seed corpus. The feature values for the n-grams are estimated from the statistics of the n-grams in the large unsegmented corpus. A set of initial weights are used to classify the word and non-word n-grams in the seed corpus according to their feature values. The weights are then adjusted according to the misclassified instances in the word or non-word n-grams until some optimization criteria for the classification results are achieved. A probabilistic descent method is used for adjusting the weights [Amari 67]. In brief, the weights are adjusted in the direction which is likely to decrease the risk, in terms of precision and recall, of the classifier.

5. Features for Classification

To classify the character n-grams, we need to use some discriminative features for the classifier. In particular, we found that the following features may be useful [Wu 93, Su 94, Tung 94].

Frequency. Intuitively, a character n-gram is likely to be a word if it appears more frequently than the average. Therefore, we use the frequency measure $f(x_i)$ as the first feature for classification.

Mutual Information. In general, a word n-gram should contain characters that are strongly associated. One possible measure to tell the strength of character association is the mutual information measure [Church 90] which had been applied successfully for measuring word association among 2-word compounds. The definition of mutual information for a bigram is defined as:

$$I(x,y) = log \frac{P(x,y)}{P(x) \times P(y)}$$

where P(x) and P(y) are the prior probabilities of the individual characters and P(x,y) is the joint probability for the two characters to appear in the same 2-gram. This measure is an indicator between the probability for the individual characters to occur independently (denominator) and the probability for the characters to appear dependently (nominator). If the mutual information measure is much larger than 0, then it tends to have strong association. To deal with n-grams with n greater than 2, such idea of dependent vs. independent was extended to the following definition for the 3-gram mutual information:

$$I(x,y,z) = log \frac{P_D(x,y,z)}{P_I(x,y,z)} = log \frac{P(x,y,z)}{P_I(x,y,z)}$$
$$P_I = P(x)P(y)P(z) + P(x)P(y,z) + P(x,y)P(z)$$

In the above definition, the nominator P_D means the probability for the three characters to occur dependently (i.e., the probability for the three characters to form a 3-character word), and the denominator P_I means the total probability (or average probability, to a scaling factor of 3) for the three characters to appear in the same 3-gram independently (i.e., by chance, possibly from two or three individual words). The extension could be made to other n-grams in a similar way.

Entropy. It is also desirable to know how the neighboring characters for an n-gram is distributed. If the distribution of the neighboring characters is random, it may suggest that the n-gram has a natural break at the n-gram boundary, and thus suggest that the n-gram is a potential word. Therefore, we use the left entropy H_L and right entropy H_R of an n-gram as another feature for classification. The left and right entropy measures are defined as follows [Tung 94]:

$$H_L(x) = -\sum_{c_i} P_L(c_i;x) \, log \, P_L(c_i;x)$$
$$H_R(x) = -\sum_{c_i} P_R(x;c_i) \, log \, P_R(x;c_i)$$

where $P_L(c_i;x)$ are the probabilities of the left neighboring characters of the n-gram x, and $P_R(x;c_i)$ are the probabilities of the right neighboring characters. It is possible to use any function of the left and right entropies for the classification task. In this paper, the average of the left and right entropies is used as a feature.

Furthermore, since the dynamic ranges of the frequencies and mutual information are very large, we used the log-scaled frequency, log-scaled mutual information and unscaled entropy measure as the features for the two class classifier. Without confusion, we will still use the terms of frequency and mutual information throughout the paper. In other words, the score vector for the classifier is

$\vec{x}_s = [\log(f), \log(I), H, 1]$. (The constant '1' is used for training an appropriate threshold.)

6. Automatic Lexical Tagging: Viterbi Training for POS Tags (VTT)

Once a word-segmented text corpus is acquired, the segmented version can be annotated with parts of speech so as to extract a POS annotated electronic dictionary. The problem of POS tagging can be formulated as the problem of finding the best possible tagging pattern that maximizes the following lexical score [Church 88, Lin 92]:

$$\begin{aligned} S_{lex} &= P\left(T_j | W\right) = P\left(t_1^n | w_1^n\right) \\ &= P\left(W | t_1^n\right) \times P\left(t_1^n\right) / P(W) \\ &\cong 1/P(W) \times \prod P\left(t_i | t_{i-1}\right) P\left(w_i | t_i\right) \end{aligned}$$

where T_j is the j-th possible set of lexical tags (parts of speech) for the segmentation pattern W. The tagging process can thus be optimized based on the product of the POS tag transition probabilities $P\left(t_i | t_{i-1}\right)$ and the distribution for $P\left(w_i | t_i\right)$. The Viterbi training process for POS tagging based on this optimization function is shown in Figure 3.

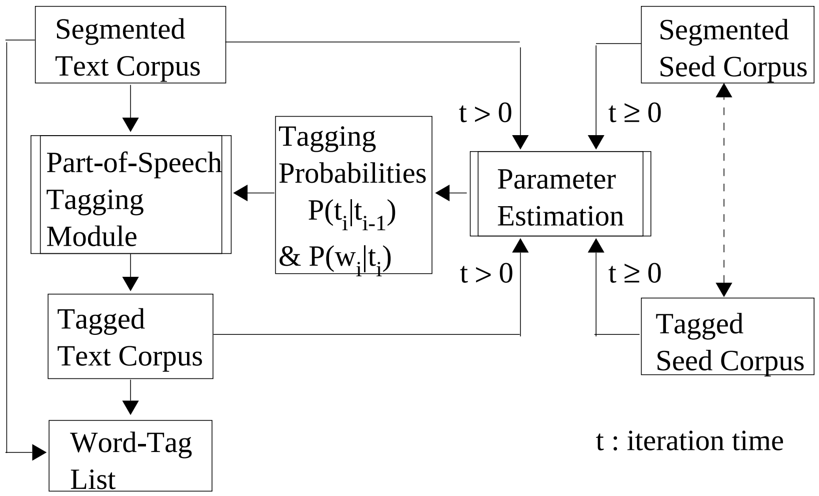

Figure 3 Block Diagram for a Viterbi POS Tag Training System

Initially, $P\left(t_i | t_{i-1}\right)$ and $P\left(w_i | t_i\right)$ are estimated from the small seed corpus. Furthermore, each n-gram in the segmented text corpus will be assigned the most frequently encountered N POS tags in the seed corpus; in our experiments, N is selected as 10 since the most frequently used 10 POS tags already cover over 90% of the tags in the seed.

During the training sessions, the various parts of speech sequences for the untagged text corpus are expanded first, and the lexical score for each path is evaluated. We then choose the path with the highest score and the corresponding parts of speech of the path for re-estimating the required probabilities. The re-estimated probabilities are acquired from both the seed corpus and the highest-scored tagging results. This process repeats until the tagging results no more change or until a maximum number of iteration is reached.

7. Integrated Systems for Dictionary Construction

There are several ways to combine the above techniques to form an integrated automatic dictionary construction system. The following sections describe two such possibilities. Their performances will be compared in the next chapter.

7.1 Basic Model: Viterbi Training for Words + Viterbi Training for POS Tags (VTW + VTT)

In the simplest topology, the Viterbi Training procedure for words is applied until the word segmentation parameters converge. The segmented text thus acquired (and hence the word n-grams) is then labelled with POS tags using the Viterbi Training procedure for POS tags. As mentioned in Figure 2, the n-grams are acquired from the unsegmented text corpus; n-grams that are less frequent than a lower bound (LB) are filtered out. The remaining n-grams then form the word candidates for expanding the various segmentation patterns.

7.2 PostFiltering Model: Viterbi Training for Words + Two-Class Classifier PostFiltering + Viterbi Training for Tags (VTW + TCC + VTT)

In the Basic Model, all n-grams that occur more frequent than 5 times in the large text corpus are considered potential words. Therefore, the number of possible segmentation patterns is extremely large. In fact, however, only about 17% of bigrams, 3% of trigrams and 4% of 4-grams in the frequency-filtered word candidates are recognized as words in a human constructed dictionary of more than 80K entries. Therefore, it is very difficult to find the best segmentation patterns, and thus the word list, with the Basic Model. To relieve the problem, the VTW module can be considered as a filter to the frequency-filtered word candidates, and we can further filter out inappropriate candidates by a TCC postfilter at the output end of the Basic Model. Intuitively, the post-TCC module will have a better chance to find out real word candidates from the output word list of the Basic Model, even though the VTW module may not perform well. The configuration is shown in Figure 4.

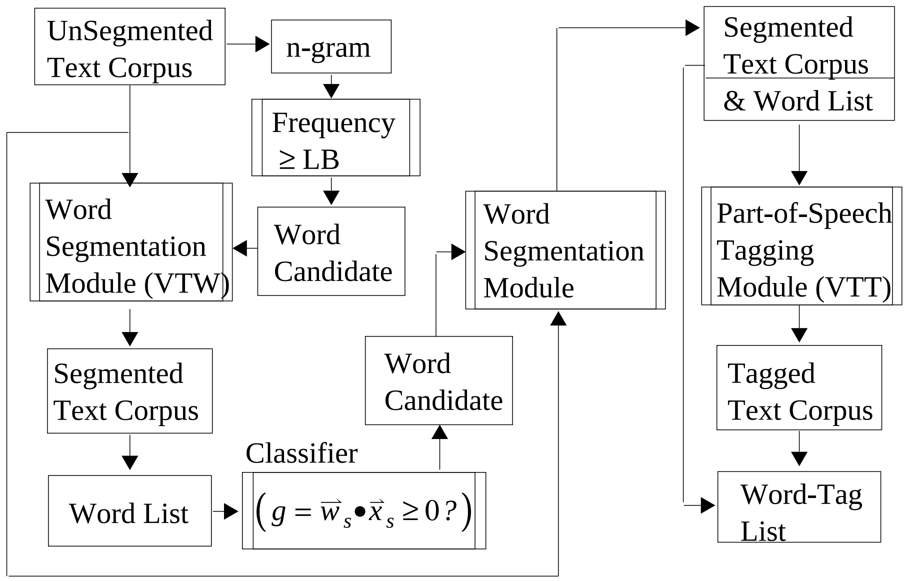

Figure 4 VTW + TCC + VTT Configuration for Automatic Construction of an Electronic Dictionary

In this topology, the Viterbi training procedure for words is applied first to acquire the possible word list which maximizes the likelihood of the segmentation patterns. The two-class classifier is then used as a postfilter to confirm whether the candidates are real word n-grams. The word n-grams thus acquired are then used as the word candidates of a second word segmentation module to produce a segmented text corpus. The segmented version is then labelled with POS tags using the Viterbi Training procedure for POS tagging.

8. Experiment Environments

In our experiments, the untagged Chinese text corpus contains 311,591sentences (about 1,670,000 words, 9 M bytes). Its major domain is news articles and reports from the China Times daily news. There are 246,036 distinct n-grams in this corpus, including 3,994 1-grams, 99,407 2-grams, 99,211 3-grams and 43,424 4-grams. Since most Chinese words are not longer than 4 characters, only 1-, 2-, 3- and 4-grams are in the word candidate list.

A seed corpus of 9,676 sentences (127,052 words, about 415 K bytes) of computer domain is available. A smaller seed of 1,000 sentences is uniformly sampled from the above corpus. This small seed corpus contains 12,849 words (about 42K bytes). The numbers of n-grams for n=1, 2, 3, 4 are 893, 7782, 12289 and 12989, respectively. Among these n-grams, only 1275 bigrams, 317 trigrams and 40 4-grams are registered as words in a dictionary.

Note that, since the numbers of word n-grams for n=3 and 4 are very small, the parameters (and performances) estimated based on such n-grams will introduce large estimation errors. Hence, the estimated performance will be very unreliable. For this reason, the conclusions will be drawn from the 2-gram performances; the performances for 3-gram and 4-gram will be listed for reference only.

9. Performance Evaluation

To get an estimation of the system performance automatically, the extracted dictionary is compared against a manually constructed *standard dictionary*. This is required because the extracted dictionary is large, and human verification will be both subjective and time-consuming. The performance will be evaluated in terms of the word precision rate and recall rate for the VTW and the TCC modules. The word precision rate is the number of n-grams common to the extracted word list and the standard dictionary divided by the number of n-grams in the extracted word list; on the contrary, the recall is the number of common n-grams divided by the number of n-grams in the standard dictionary. The VTT module will be estimated in terms of several weighted tag precision and recall rate measures.

The standard Word Dictionary to be compared with the extracted word list is acquired by merging the word lists of two electronically available dictionaries [CKIP 90, BDC 93] and the words included in the seed corpus. It also excludes all n-grams which never appear in the 9767-sentence seed corpus and the untagged text corpus, because such n-grams will never be the input to the dictionary construction system. The merged dictionary, excluding entries that appear less frequently than the frequency lower bound (5), contains 17,005 bigram words, 2,524 trigram words and 1,612 4-gram words.

The standard Word-Tag Dictionary to be compared with the extracted POSes is constructed from the BDC English-Chinese electronic dictionary [BDC 93]. The derived Word-Tag Dictionary contains 87,551 entries, including 35,722 bigram words, 19,858 trigram words, and 24,092 4-gram words. The tagset used in this dictionary contains 62 tags (including two punctuation tags). Note that there are only 42 tags in the smaller seed corpus of 1000 sentences, and the whole seed corpus of 9676 sentences contains only 47 POS tags (including one punctuation tag). Therefore, such missing tags will introduce some tag extracting errors in the training processes.

Since the Word Dictionary and Word-Tag Dictionary, which are used for comparison with the extracted dictionary, are constructed independently of the corpus from which the lexicon entries are extracted, the reported performances could be greatly *underestimated*. For instance, an n-gram which is identified as a lexicon entry by the system but excluded from the Word Dictionary may not necessarily be a wrong word entry if it is judged by an expert lexicographer. In the ideal case, the Word Dictionary and Word-Tag Dictionary should be constructed by an expert lexicographer based on the corpus for a fair comparison. Unfortunately, we are unable to afford the man power for such an evaluation on the large corpus. Therefore, special attention should be taken when interpreting the performances reported in the following sections.

9.1 Performance for the Basic (VTW+VTT) Topology

Table 1 shows the performances in different stages for the Basic Model (columns 1-4) and the Postfiltering Model (columns 1-6) by using the small (1000-sentence) seed corpus. (Columns 1-4 are shared because the Postfiltering is applied immediately after the Basic Model.) The numerators in the parentheses are the numbers of correctly identified n-grams; for precision, the denominators are the numbers of n-grams in the extracted word lists; and for recall, they stand for the numbers of n-grams in the standard dictionary.

The third column simply shows the initial precision and recall for the n-grams which are more frequent than a frequency lower bound LB; such word candidates are the base for evaluating the effects of the VTW and TCC modules. The Viterbi training process for extracting the word list goes through 4 iterations. With the small seed corpus, it is observed that the precision for bigram words is improved from the initial precision of 17.07% to 38.21%, corresponding to an increase of 21.14%, and the recall is dropped from 100% to 89.87%, a decrease of 10.13%. This shows that the Viterbi training procedure does provide a significant improvement in precision while maintaining a reasonable recall.

Note that, the precision for the initial (frequency-filterred) word candidates with respect to the dictionary is an indicator to the difficulty of the task. It indicates how much percentage of word candidates are recognized as words by the standard dictionary. From the table, the initial word candidates in the large corpus only include 3 to 4 % of the real word candidates which are recognized as words by a human constructed dictionary. Furthermore, there are only 317 trigram words and 40 4-gram words in the training seed corpus. As a result, it is difficult to spot such candidates from the large candidate list with a reasonable precision and recall. Hence, it is not surprising that the performance for the 3-grams and 4-grams is poor. For these reasons, we will make no further comments on the 3-gram and 4-gram performances which are trained and observed under a very difficult training environment. A few comments will be given on the section for error analysis though.

n-gram	Processing Step	Freq. LB. Filtering (LB=5)	VTW	Two-Class Classifier PostFiltering (TCC)	VTW-2
2	Precision	17.07 (17,005/ 99,601)	38.21 (15,283/ 39,999)	56.80 (13,091/23,049)	56.88 (13,156/23,130)
	Recall	100.0 (17,005/ 17,005)	89.87 (15,283/ 17,005)	76.98 (13,091/17,005)	77.37 (13,156/17,005)
3	Precision	2.54 (2,524/99,460)	6.01 (2,171/36,123)	6.02 (2,171/36,067)	6.12 (2,170/35,443)
	Recall	100.0 (2,524/2,524)	86.01 (2,171/ 2,524)	86.01 (2,171/2,524)	85.97 (2,170/2,524)
4	Precision	3.71 (1,612/43,454)	5.81 (1,503/25,891)	6.21 (1,497/24,099)	6.31 (1,497/23,713)
	Recall	100.0 (1,612/1,612)	93.24 (1,503/1,612)	92.87 (1,497/1,612)	92.87 (1,497/1,612)

Table 1. Word Identification Performance for the VTW+VTT and VTW+TCC+VTT topologies (seed=1000 sentences)

9.2 Performance for the Postfiltering (VTW+TCC+VTT) Topology

The performance of the PostFiltering model is shown in columns 1-6 of Table 1. The two VTW modules in Figure 4 are identical, and each VTW module goes through 4 training iterations. With the small seed corpus, the bigram performance is improved from 38.21% to 56.80% with a decrease of recall from 89.87% to 76.98% after the post-filter is installed. The global system achieves a precision rate of 56.88% at the recall rate of 77.37%.

It is observed that, by using the large corpus (which is about ten folds in size), the precisions are only slightly increased (by about 2%). Therefore, the corpus size may not be a critical issue in this task. A better extraction model might be more likely to improve the system further.

9.3 Error Analysis for Word Identification Models

The 3-gram and 4-gram precision rates are quite poor in the above tests. An inspection of the entries which are not recognized as words shows that some of the entries which should be considered words are not registered in the standard general dictionary. This means that the system does find some *new words* that were never seen by the standard dictionary, and thus are considered wrong. Examples of such n-grams are:

互惠原則、內閣閣員、天佑證券、仁警分局、卜蜂集團、少量多樣、心路歷程、戶口謄本、毋枉毋縱、乙太網路、 分析師、分隊長、夫妻檔、月成長、水源區 (...)。

Some of the above examples are frequently encountered domain-specific terms in politics, economics, etc., which would be considered new words to a general dictionary. Others include frequently encountered proper names (company names, city names) or productive lexicon entries. Although such terms may not be considered in constructing a general dictionary, it is useful to include such daily used high frequency terms in an electronic dictionary for practical processing purposes. Therefore, the precision performance, estimated by comparing it with a general dictionary, is usually underestimated.

Excluding such n-grams, the other incorrectly extracted n-grams have some special patterns which suggest that the extraction models might be refined by extracting or filtering out n-grams according to the substring patterns they have. In particular, a 3- gram (or 4-gram) may have the following relationships with its substrings:

1. compositional: the n-gram can be decomposed into legal words (e.g., 今天下午 ("this afternoon")、今天發表 ("announce ... today")、介入選舉("intervene the election")).

2. collocational: parts of the n-gram are legal words, the other parts are highly flexible (e.g.,"do not + VERBS"：不予起訴、不予採納、不予理會；"many + NOUNS"：不少民眾、不少台商、不少住戶；"not + ADJECTIVES"：不公平、不切實、不友善).

3. idiomatic: none of the substrings are legal words, all single characters are highly flexible (e.g., 不一而足 ("cannot be enumerated one-by-one")).

All the above patterns are related to the internal structure of the n-grams; our features and models, however, are more closely related to the intrinsic properties of the n-gram itself or the contextual information with the other n-grams. This explains why some highly associated n-grams, which are not word units, are extracted as words by the system. It also suggests that we could filter out some inappropriate candidates which contain frequently encountered substrings and whose other parts show high entropy (or similar measures.) A few simple filtering rules based on such observation show that the precision could be increased more effectively by refining the models in this way than increasing the seed corpus size. A more extensive survey is being studied.

9.4 Tagging Accuracy: Weighted Tagging Recall and Precision

Because a word may be tagged differently under different context, a word identified by the VTW or TCC module may have more than one tag. For the tagging accuracy, we use several measures to estimate the performance. Firstly, the number of word-tag pairs common to the extracted word-tag list and the Word-Tag Dictionary divided by the number of pairs in the extracted list is defined as the *raw precision rate*; the *raw recall rate* is defined similarly as the number of common word-tag pairs divided by the number of word-tag pairs in the Word-Tag Dictionary. With this measure, if a word in the extracted list has M tags, then all the M word-tag pairs for the word are evaluated independently of the other pairs.

Because the annotated tags for a word is usually considered as a whole when constructing a dictionary entry, it may be desirable to define a per-word precision and per-word recall to measure how good the tags for a word is annotated, and then properly associate a weight to each word to evaluate the performance for the whole system.

The per-word precision for a word is defined as the number of tags commonly annotated in the dictionary entry and the extracted word-tag list for the word divided by the number of tags in the extracted word-tag entry for the word. On the contrary, the number of common tags divided by the number of tags in the corresponding dictionary entry is defined as the per-word recall for the word. For instance, if a word is tagged with the parts of speech [n, v, a] by the system, and it has the parts of speech [n, adv] in the standard dictionary, then the per-word recall will be 1/2 for this word and the per-word precision will be 1/3.

Based on the per-word precision and recall, we define the *average precision* (resp. *recall*) of the system as the sum of per-word precisions (resp. recalls) divided by the number of words in the word list. Alternatively, we could take the frequencies of the n-grams into account so that more frequently used words are given a heavier weight on its per-word precision and recall. Such weighted precision (or recall) is defined as the sum of product of the per-word precision (or recall) and the word probability taken over each word.

9.5 Part-of-Speech Extraction Performance

To evaluate the performance of the Viterbi Part-of-Speech Tagging Module on the POS extraction task, the words in the segmented and POS tagged text corpus are compared against the Word-Tag Dictionary mentioned in a previous section.

Since not all extracted words have a corresponding entry in the Word-Tag Dictionary, we only evaluate the performance of the POS extraction module over common entries in both the extracted dictionary and the standard dictionary. The sizes of the common entries for the various models are around 8 to 9 thousands entries. On the average, each dictionary entry contains about 1.4 parts of speech, and each entry annotated by the Viterbi training module has about 1.7 parts of speech.

Tables 2 shows the raw precision (Praw), average precision (Pavrg), weighted precision (Pwavg), and their corresponding recall rates. (The left-hand side performance is acquired with a seed of 1000 sentences, and the right hand side with 9676 sentences.) It seems that the performance is not significantly different between the two different models. This may imply that the segmented text corpus passed from the various models do not have significant difference. Furthermore, unlike in the word identification stage, the increase in seed size does provide significant improvement on precision and recall. With the large seed corpus, the weighted precision and recall are 71% and 73%. Considering the fact that the parts of speech are optimized from 10 parts of speech for each word, the results are reasonably acceptable.

	Basic Model	Post-Filtering	Basic Model	Post-Filtering
Praw	46.40 (7944/17119)	46.37 (7241/15615)	51.79 (8873/17132)	52.53 (8390/15973)
Rraw	60.40 (7944/13153)	60.82 (7241/11906)	64.22 (8873/13816)	64.61 (8390/12986)
Pavrg	53.07	53.17	60.21	61.25
Ravrg	68.69	69.54	72.79	73.59
Pwavrg	57.20	57.55	71.16	71.81
Rwavrg	71.29	71.58	73.42	73.83

Table 2. Performance for Part-of-Speech Extraction of the Two Models
(Seed=1000 and Seed = 9676, respectively)

10. Concluding Remarks

In this paper, we propose an unsupervised reestimation approach and a two-class classification method to extract embedded words from a large unsegmented Chinese text, and assign possible parts of speech to each word with a similar reestimation method. An electronic dictionary with parts of speech information can thus be acquired automatically.

It is observed that the system could acquire POS-tagged lexicon entries with a reasonably acceptable precision and recall. Since this approach adopts an unsupervised learning approach to construct the dictionary, its performance, in terms of precision and recall, is less satisfactory than a supervised learning strategy, where a large tagged corpus and dictionary are used. However, it requires little human intervention in the whole process, the cost to construct the dictionary, in terms of budget and time for pre-tagging, is much smaller than a supervised learning approach. Therefore, it is worth while trading off the precision requirement with the cost of dictionary construction. With the results of this preliminary study, it is expected that the current techniques described here could form a good basis for constructing a better and automatic dictionary construction system.

References

[Amari 67] Amari, Shunichi, "A Theory of Adaptive Pattern Classifiers," *IEEE Trans. on Electronic Computers*, Vol. EC-16, No. 3, pp. 299-307, 1967.

[BDC 93] Behavior Design Corporation, "The BDC Chinese-English Electronic Dictionary: Version 2," Hsinchu, Taiwan, ROC, 1993.

[Chang 91] Chang, Jyun-Sheng, C.-D. Chen and S.-D. Chen, "Chinese Word Segmentation through Constraint Satisfaction and Statistical Optimization," (in Chinese) Proceedings of ROCLING-IV, ROC Computational Linguistics Conferences, pp. 147--165, National Chiao-Tung University, Hsinchu, Taiwan, ROC, 1991.

[Chang 93] Chang, Chao-Huang and Cheng-Der Chen, "HMM-based Part-of-Speech Tagging for Chinese Corpora," *Proceedings of the Workshop on Very Large Corpora*, WVLC-1, pp. 40-47, Ohio State University, 1993.

[Chen 86] Chen, K.-J., C.-J. Chen and L.-J. Lee, "Analysis and Research in Chinese Sentence Segmentation and Construction," Technical Report, TR-86--004, Taipei: Academia Sinica, 1986.

[Chiang 92] Chiang, T.-H., J.-S. Chang, M.-Y. Lin and K.-Y. Su, "Statistical Models for Word Segmentation and

Unknown Word Resolution", *Proceedings of ROCLING V*, pp. 121-146, National Taiwan University, Taiwan, ROC, 1992.

[Church 88] Church, K., "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *ACL Proc. 2nd Conf. on Applied Natural Language Processing*, pp. 136-143, Austin, Texas, USA, 9-12 Feb. 1988.

[Church 90] Church, K. and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, vol. 16, pp. 22-29, Mar. 1990.

[CKIP 90] Chinese Knowledge Information Processing Group, "The CKIP Electronic Dictionary," Academia Sinica, Taipei, Taiwan, ROC, 1990.

[Dempster 77] Dempster, A. P., N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, 39(B), pp. 1-38, 1977.

[Fan 88] Fan, C.-K. and W.-H. Tsai, "Automatic Word Identification in Chinese Sentences by the Relaxation Technique," *Computer Processing of Chinese and Oriental Languages*, vol. 4, no. 1, pp. 33--56, 1988.

[Ho 83] Ho, W.-H., "Automatic Recognition of Chinese Words," master thesis, National Taiwan Institute of Technology, Taipei, Taiwan, 1983.

[Lin 92] Lin, Y.-C., T.-H. Chiang and K.-Y. Su, "Discrimination Oriented Probabilistic Tagging", *Proceedings of ROCLING V*, pp. 85-96, National Taiwan University, Taiwan, ROC, 1992.

[Sproat 90] Sproat, R. and C. Shin, "A Statistical Method for Finding Word Boundaries in Chinese Text," *Computer Processing of Chinese and Oriental Languages*, vol. 4, no. 4, pp. 336--351, 1991.

[Su 94] Su, K.-Y., M.-W. Wu and J.-S. Chang, "A Corpus-based Approach to Automatic Compound Extraction," *Proceedings of ACL 94*, pp. 242-247, New Mexico State University, June, 1994.

[Tung 94] Tung, Cheng-Huang and Hsi-Jian Lee, "Identification of Unknown Words from a Corpus," *Computer Processing of Chinese & Oriental Languages*, Vol. 8, pp. 131-145, (*Proceedings of ICCPOL-94*, pp. 412-417, Taejon, Korea,) Dec. 1994.

[Wu 93] Wu, M.-W. and K.-Y. Su, "Corpus-based Automatic Compound Extraction with Mutual Information and Relative Frequency Count," *Proceedings of ROCLING VI*, pp. 207-216, Nantou, Taiwan, ROC, Sep. 1993.

[Yeh 91] Yeh, C.-L. and H.-J. Lee, "Rule-Based Word Identification for Mandarin Chinese Sentences --- A Unification Approach," *Computer Processing of Chinese and Oriental Languages*, vol. 5, no. 2, pp. 97--118, March 1991.