Computational Linguistics and Chinese Language Processing
vol. 2, no. 2, August 1997, pp. 97-148.
© Computational Linguistics Society of R.O.C.

An Unsupervised Iterative Method for Chinese New Lexicon Extraction

Jing-Shin Chang* and Keh-Yih Su+

ABSTRACT

An unsupervised iterative approach for extracting a *new lexicon* (or *unknown words*) from a Chinese text corpus is proposed in this paper. Instead of using a non-iterative segmentation-merging-filtering-and-disambiguation approach, the proposed method iteratively integrates the contextual constraints (among word candidates) and a joint character association metric to progressively improve the segmentation results of the input corpus (and thus the new word list.) An *augmented* dictionary, which includes *potential unknown words* (in addition to known words), is used to segment the input corpus, unlike traditional approaches which use only known words for segmentation. In the segmentation process, the augmented dictionary is used to impose contextual constraints over known words and *potential unknown words* within input sentences; an unsupervised Viterbi Training process is then applied to ensure that the selected potential unknown words (and known words) maximize the likelihood of the input corpus. On the other hand, the joint character association metric (which reflects the global character association characteristics across the corpus) is derived by integrating several commonly used word association metrics, such as mutual information and entropy, with a joint Gaussian mixture density function; such integration allows the filter to use multiple features simultaneously to evaluate character association, unlike traditional filters which apply multiple features independently. The proposed method then allows the contextual constraints and the joint character association metric to enhance each other; this is achieved by iteratively applying the joint association metric to truncate unlikely unknown words in the augmented dictionary and using the segmentation result to improve the estimation of the joint association metric. The refined augmented dictionary and improved estimation are then used in the next iteration to acquire better segmentation and carry out more reliable filtering.

Experiments show that *both* the precision and recall rates are improved almost *monotonically*, in contrast to non-iterative segmentation-merging-filtering-and-disambiguation approaches, which often sacrifice precision for recall or *vice versa*.

* Department of Electrical Engineering, National Tsing-Hua University, Hsinchu,Taiwan, ROC.
e-mail: shin@hermes.ee.nthu.edu.tw

+ Behavior Design Corporation, 2F, No. 5, Industrial East Rd IV, Science-Based Industrial Park, Hsinchu,Taiwan,ROC.

With a corpus of 311,591 sentences, the performance is 76% (bigram), 54% (trigram), and 70% (quadragram) in *F-measure*, which is significantly better than using the non-iterative approach with F-measures of 74% (bigram), 46% (trigram), and 58% (quadragram).

Keywords: Unknown Word Identification, New Lexicon Extraction, Unsupervised Method, Iterative Enhancement, Chinese, Lexicon

1. Introduction

A large-scale electronic dictionary is the fundamental component of many natural language processing applications, such as spell checking and machine translation. However, *new words* (or *unknown words*, as defined in [Wang 95], including new compound words) are appearing continuously in various domains, especially with the rapid growth of the Internet community. Quickly acquiring *new* words that are not registered in an existing dictionary is, thus, very important.

For instance, a machine translation system for translating computer manuals may need to update its lexicon frequently to keep up with the constantly changing computer technologies because the translation of many newly generated compound words is not compositional in terms of known words inexisting dictionaries.

Furthermore, from our experience in running the BehaviorTran machine translation system [Chen 91], the number of new lexical entries may exceed several thousands, especially for large translation projects. Under such circumstances, manually scanning a corpus to extract all the new words will be costly and time-consuming. In addition, it is difficult for lexicographers to judge objectively which new words should be included into a lexicon if certain quantitative indices are not provided. Therefore, an automatic method for new lexicon acquisition is important for adapting a dictionary promptly to our quick changing world, with little cost and high coverage in different domains.

The above requirements apply to both the English and Chinese languages. However, new Chinese word extraction is more difficult since there are no natural delimiters, like spaces, between Chinese words. Hence, an unsupervised approach capable of segmenting a large text corpus to extract new words is desirable in compiling a large Chinese dictionary.

A few closely related works [Chiang 92, Lin 93a, Lin 93b, Smadja 93, Wu 93, Su 94, Fung 94, Tung 94, Chang 95, Wang 95, Smadja 96] have been introduced for finding English or Chinese new words in a large corpus. The works in [Chiang 92, Lin 93a, Lin

93b, Fung 94, Tung 94, Chang 95, Wang 95], in particular, are related to Chinese unknown word identification. In this paper, we will also focus on how to extract Chinese unknown words from a text corpus.

Some of the above works on Chinese unknown word extraction ([Chiang 92, Lin 93a, 93b, Tung 94]) require a pre-annotated corpus for supervised training. For instance, [Tung 94] used a segmented corpus with part-of-speech tags to train parameter values. Since the human cost is high in preparing such a training corpus, we will focus on the *unsupervised* method for new word extraction.

Although they are not exactly the same, many previous works, such as [Tung 94, Wang 95], can be roughly characterized by the following segmentation-merging -filtering-and-disambiguation steps: (1) The Chinese text corpus is *segmented* into possible word segments by looking up an existing dictionary (hereafter, the *system dictionary*) and using a word segmentation model to select the best segmentation pattern. (2) Potential unknown word candidates are then formed by *merging* adjacent segments (i.e., known words or single characters) in the segmented corpus (since many *unknown words*, which are not in the system dictionary, will be segmented into known words and single characters after segmentation.) (3) Afterward, a set of association metrics or testing statistics, such as mutual information [Church 90], entropy [Tung 94], or an association strength measure [Smadja 93, Wang 95], are used by a *filter* to filter out candidates which have low associations. (4) Finally, an optional step is used to resolve *ambiguity* between overlapping candidates which are identified as new words at the same time [Tung 94]. For instance, ' 漁 業 區 附近 ' will produce the overlapping candidates: ' 漁業區 ' and ' 區附近 ', which need be resolved to identify which one is a stronger competitor. In [Tung 94], the entropy information, which was also used by the filter, was used in a different way to determine which overlapping candidate is the stronger one. Furthermore, the above 4 steps are usually executed only once without iteratively looping back.

1.1 Problems with Segmentation-Merging Using Known Words Only

The above-mentioned non-iterative segmentation-merging-filtering-disambiguation approaches can be easily implemented to acquire new word candidates for human post-editing; by adjusting some thresholds *via* trial-and-error, the precision or recall can also be adjusted to fit the lexicographer's needs. However, they may not be easy to tune to improve *both* the recall and precision rates *at the same time*. (See the next section on why a *filtering* approach normally cannot improve precision and recall simultaneously.) Also, in general, they have a few other problems which must be resolved.

First of all, some of the unknown words cannot be recovered by using the 'merge' operation. In fact, it was shown in [Chiang 92, Lin 93a, Lin 93b] that, when there are unknown words in a text corpus, bad segments are generated due not only to over-segmenting of unknown words into shorter segments, but also to *incorrect merging* of known words and/or single characters into longer segments [Lin 93b]. For instance, ' 土地 公有 政策 ' may be segmented incorrectly into ' 土地公 有 政策 ', in which ' 土地公 ' is an over-merged string if ' 公有 ' is an unknown word and ' 土地公 ' is already registered in the system dictionary. Although it is possible for the 'merge' operation to recover unknown words which suffer from over-segmentation errors for further justification, the unknown words (such as ' 公有 ' in the last example) which suffer from the over-merging error cannot be recovered by simply using the 'merge' operation.

Second, the filter cannot take advantage of contextual constraints on *unknown words* to produce better segmentation (and unknown word candidates) because suspected unknown words do not participate in the segmentation process. Under such circumstances, a large number of unknown word candidates (including many spurious word candidates) will be generated by the merge operation and then submitted to the filter. Many such spurious words, however, cannot be rejected by the filter since it is the contextual constraints, not the association features used by the filter, that reject such spurious words. With our corpus, for instance, such randomly merged unknown word candidates amount to more than 2 million strings (of 2-4 characters). The segmentation-and-merging process (using only known words for segmentation) may thus result in low system precision.

For example, the segmentation pattern '(獲) 省都委會 同意 ' ('gain the approval of the Provincial City Development Committee') is identified by our lexicographer as the preferred segmentation. It will be segmented correctly if the (abbreviational) unknown word ' 省都委會 ' is included in the system dictionary and, thus, participates in the segmentation process. In this case, no spurious word candidates will be submitted to the filter. On the other hand, if we only use known words for segmentation, the above string will be segmented into '(獲) 省 都 委 會同 意 '. Also, by merging the short segments, we will have the spurious word candidates ' 省都 ' ('capital of the Province'), ' 省都委 ' ('committee members of the Provincial City Development Committee'), ' 都 委 ' ('committee members of the City Development Committee'), ' 都 委會同 ' ('the committee members of the City Development Committee call a meeting to...'), ' 委會同 ', ' 委會同意 ', and ' 會同意 ' ('... will approve'); many such spurious word candidates will be accepted by the filter as legal words (since they might be highly associated). However, they should actually not be extracted in the current context if we agree that '省都委會' and '同意' are the only and the most preferred lexical entries in the above phrase.

The segmentation-and-merging scheme will thus degrade the precision performance of the system due to the introduction of a large number of spurious words. Such spurious words, however, will not be generated if highly likely potential unknown word candidates, such as '省都委會', are added to the system dictionary and participate in the segmentation process. In this case, only those unknown word candidates that are preferred by the segmentation process will survive and be submitted to the filter for further justification.

Third, a separated ambiguity resolution step might be required, due to the merging step, to remove overlapping ambiguities, such as the previous '漁業區' and '區附近' example. Normally, the disambiguation step only compares adjacent candidates in a local context to decide which candidate will survive without using contextual constraints over the whole sentence to see whether the resultant segmentation is the desired segmentation pattern. In other words, such a disambiguation process does not choose candidates to maximize the overall likelihood of the corpus. Therefore, this extra disambiguation process may incorrectly disambiguate some overlapping ambiguities which need be resolved using the contextual information and, thus, may degrade the system performance.

For instance, '彰化 縣警 刑警隊 少年組' may be segmented into '彰化 縣 警 刑警隊 少年組' due to the lack of the unregistered new word '縣警' ('county police office'). After merging the segments, the two overlapping candidates '彰化縣' and '縣警' may both be qualified by the filter as words, but the ambiguity resolution step may reject '縣警' as a new word since it is a weaker competitor than '彰化縣' in the corpus. However, although '縣警' might be a weaker competitor than '彰化縣', it is very unlikely that '彰化縣 警 刑警隊 少年組' is a good segmentation pattern because the single character '警' is rarely used as a word by itself. In other words, if we know that the joint likelihood for '彰化縣' and '警' (in terms of the product of the their individual probabilities $P(\text{彰化縣}) \cdot P(\text{警})$) is much smaller than the joint likelihood of '彰化' and '縣警' (with likelihood $P(\text{彰化}) \cdot P(\text{縣警})$), we should not discard the new word '縣警' even though it is a weaker competitor than its neighbor. This means that such a (merging-)filtering-and-disambiguation process may prefer strong competitors regardless of the context. If, on the other hand, the potential unknown word candidates '彰化縣' and '縣警' are both included in the system dictionary, then the additional disambiguation process will be unnecessary. The above problem can then be avoided.

In summary, the above problems can be overcome if an *augmented* dictionary, which contains *potential unknown words* (in addition to known words), can be used during the segmentation process to achieve a better segmentation. Under such circumstances, the improved segmentation result will contain highly likely unknown word candidates. Accordingly, the precision of the filter will benefit from the better segmentation generated by using such an enlarged dictionary. For this purpose, an *augmented dictionary*, which includes both *potential unknown words* in the input corpus and known words in the system dictionary, will be used in our system in the segmentation process; and the probabilities associated with such potential unknown words will be jointly trained with other known words to globally maximizes the likelihood of the input text. Thus, the separated merging and disambiguation processes in conventional approaches will no more be necessary.

1.2 Problems with the Non-Iterative Scheme

Although including potential unknown words in the segmentation process resolves the above mentioned problems of the segmentatiom-merging-filtering-disambiguation scheme, segmentation and filtering are still two independent steps; thus, they cannot enhance one-another's performance simply by being cascaded.

To address this issue, we must first note that the performance of the segmentation module greatly depends on whether the augmented dictionary is close to the ideal dictionary (which contains *all and only* those words in the input corpus.) On the other hand, the performance of the filter depends greatly on whether its model is close to the true (lexicographer's) model and whether the model parameters are reliably estimated. We can, therefore, improve system performance by improving the augmented dictionary and/or improving the parameters of the classifier.

Initially, it is obviously impossible to construct an augmented dictionary that contains all and only those words in the input corpus. Therefore, it is not surprising that some of the unknown word candidates identified by the segmentation module will still be spurious although many of them might already be correct unknown words. In addition, it is impossible, initially, to estimate the model parameters of the filter reliably since the filter usually makes judgements based on its statistical model about words and non-word n-grams, but we are not sure which n-grams are words and which are not, except for the words in the system dictionary.

However, we can use the filter to remove spurious words from the augmented dictionary and thus prevent them from appearing in the best segmentation pattern. This is possible since such spurious unknown word candidates, which are qualified by the word segmentation process simply because they co-occur frequently, may be rejected by the filter since they may not be strongly associated according to the criteria of the filter. (For instance, as given in Section 4.2, the filter may use a normalized co-occurrence frequency, instead of the co-occurrence frequency alone, to evaluate the strength of association). On the other hand, we can also use a well-segmented corpus to help in estimating the parameters of the statistical model better (for instance, by moving those highly likely unknown words from the non-word class to the word class so that our statistical knowledge about the distributions of the word/non-word classes is better justified). Therefore, to improve the system performance further, we can form a feedback path to refine the augmented dictionary using the association features of the filter and refine the model parameters of the filter based on the segmentation results. This process can then be applied *iteratively* to enhance the individual performance of the two modules and, thus, the preformance of the whole system.

For example, the phrase '(移送) 台中少年法庭審理 ' ('(send ... to) the Taichung Youth-Court for investigation') will be segmented as '(移送) 台中少年 法庭審理 ' ('send the young boy who lives in Taichung to the court for investigation') initially due to the high frequency of the potential unknown words ' 台中少年 ' and ' 法庭審理 '. If we use the association metrics of the filter to remove these two spurious words from the augmented dictionary, the segmentation results will be progressively refined. In fact, by removing the first segment ' 台中少年 ' from the augmented dictionary using word association metrics, the best segmentation becomes '(移送) 台中 少年 法庭審理 '; this refined segmentation pattern allows us to better re-estimate the model parameters of the filter and, thus, results in further deletion of the string ' 法庭審理 ' from the augmented dictionary; the deletion of ' 法庭審理 ' finally gives us the correct segmentation '(移送) 台中 少年法庭 審理 '. This process not only improves the precision (due to the filtering operations), but also improves the recall (by applying additional segmentation sessions iteratively.)

Therefore, an *iterative* scheme is proposed here to fully integrate the contextual information used by the segmentation module and the association features used by the filter. In this iteratively integrated scheme, the filter is used to rank potential unknown words such that very unlikely potential unknown words can be removed from the augmented dictionary, and the parameters of the filter are improved according to the better and better estimated word population statistics acquired from better segmentation results.

With such an iterative scheme, the segmentation output is expected to be improved continuously through use of the progressively refined augmented dictionary, which is acquired by using the association information to filter out inappropriate candidates. By iteratively refining the augmented dictionary and, thus, the segmentation output, we can also tune the filter model and its parameters continuously using the progressively refined segmentation output.

There is one important point that is worth mentioning here. It is possible simply to reject the spurious words (such as '台中少年' and '法庭審理') by using other association features and, thus, improve the *precision* rate, as conventional non-iterative filtering approaches do. However, simply rejecting such candidates won't tell us what they really should be and, thus, won't be helpful for improving the *recall* rate. More precisely, successful filtering will only improve the precision performance, *not* the *recall* rate; however, unsuccessful filtering will degrade both the precision and recall. None of these filtering operations will improve the *recall* rate. Hence, conventional non-iterative filtering approaches usually cannot improve the precision and recall simultaneously. In contrast, the iterative scheme proposed here provides a way to recall real new words (converted from some truncated spurious words) in later segmentation processes. In this way, the *precision* will be improved by *filtering*, and the *recall* will be improved by *re-segmentation*. We can then expect to improve *both* precision and recall in the iterative scheme without sacrificing precision for recall or *vice versa*. These advantages are unlikely to be fully utilized in a non-iterative scheme.

To sum up, the general non-iterative segmentation-merging-filtering-disambiguation scheme is incapable of recovering over-merging type segmentation errors; the merging operation may also introduce many randomly merged segments to the filter, and a separate disambiguation process might be needed to resolve overlapping ambiguities. With the extra disambiguation process, the system performance might be degraded by some ambiguous pairs which can be resolved only by using contextual information. Furthermore, due to the non-iterative nature of this scheme, the segmentation module and the filter module cannot help each other to acquire better performance; most likely, precision and recall will not be improved simultaneously if the two modules are simply cascaded since no feedback path is provided to recall the real unknown words corresponding to the rejected candidates.

To resolve such problems, an augmented dictionary, which contains potential unknown words, is used in the segmentation process. The augmented dictionary of the segmenter is refined by the filter. The model parameters of the filter are also progressively refined by using the word and non-word knowledge acquired from the progressively refined word segmentation output. Such progressive refinement is conducted through an iterative scheme to re-segment the input corpus and re-estimate the filter parameters. In this iterative process, the precision is improved by truncating inappropriate candidates in the augmented dictionary, and the segmentation process provides a way to recall real new words for such truncated candidates; *both* precision and recall are, thus, improved progressively.

1.3 The Filter Design Problems

Besides problems with the conventional non-iterative scheme, there are also problems with the design of the filter. In particular, the features used by the filter, such as mutual information [Church 90], entropy [Tung 94], association strength [Smadja 93, Wang 95] and dice coefficients [Salton 83, Smadja 96], are usually applied independently, and the best sequence to apply such features for filtering is usually not known. Furthermore, the thresholds for such association metrics (to be used in qualifying the unknown word candidates) are usually set heuristically (or empirically) to get either high recall or high precision (but often not both) for a particular domain. As a result, such values must be decided by trial-and-error if the distribution of the (unknown) words is changed in different time or application domains. And the performance of the system will heavily depend on the thresholds.

To overcome these problems, a Likelihood Ratio Ranking Module (LRRM), based on a two-class minimum error classification model [Su 94], is used as our `filter' to combine all the available association metrics into one *joint* association measure, namely the log-likelihood ratio (LLR), instead of using different filters and heuristic thresholds to independently apply the word association metrics one-by-one; moreover, such a measure is used to rank the candidates in terms of the degree of association, so that we can tell which candidates are more likely (or more unlikely) to be words with respect to other candidates (instead of asserting which candidates are qualified words by checking their association metrics against some thresholds).

With such a relative ranking index, only a small fraction (say 5%) of the most unlikely unknown word candidates are truncated from the augmented dictionary, and only a small fraction of the most likely unknown word candidates are used to update the filter parameters. In other words, the filter is not using an absolute threshold value for filtering; instead, a *relative* mode for filtering is adopted. Therefore, the proposed method will not heavily depend on the determination of an 'optimal' threshold (in whatever sense) to improve *both* the precision and recall rates, in contrast to other systems which heavily depend on thresholds for tuning the precision or recall rate. This strategy is particularly useful for the current task since we are operating in an unsupervised mode of operation, in which the classifier parameters may not be reliably estimated. Because the 'classifier' now serves simply as a ranking device to supplement the segmentation module, the unknown word extraction task is modeled mainly as an iterative segmentation task, which uses an iteratively refined *augmented dictionary* in the segmentation process.

In the following sections, a brief system overview and the assessment method are presented first. We then discuss how unknown words introduce segmentation errors and how such a problem can be improved by jointly including potential unknown words during the segmentation process. Techniques for refining the list of potential unknown words and the filter (i.e., LRRM) parameters are then addressed, so that we can iteratively improve the segmentation results and the filter parameters, and thus the output (unknown) word list. (In this paper, we will sometimes use the terms 'filter', 'likelihood ratio ranking module' and 'two-class classifier' interchangeably for convenience.) Performance evaluation is also conducted to estimate the system performance, with some significant tests to ensure that the improvement is statistically significant. Furthermore, quantitative analyses are given to justify our strategies, and segmentation errors are analyzed to find possible features for our future works.

2. System Overview and Assessment Procedure

2.1 System Overview

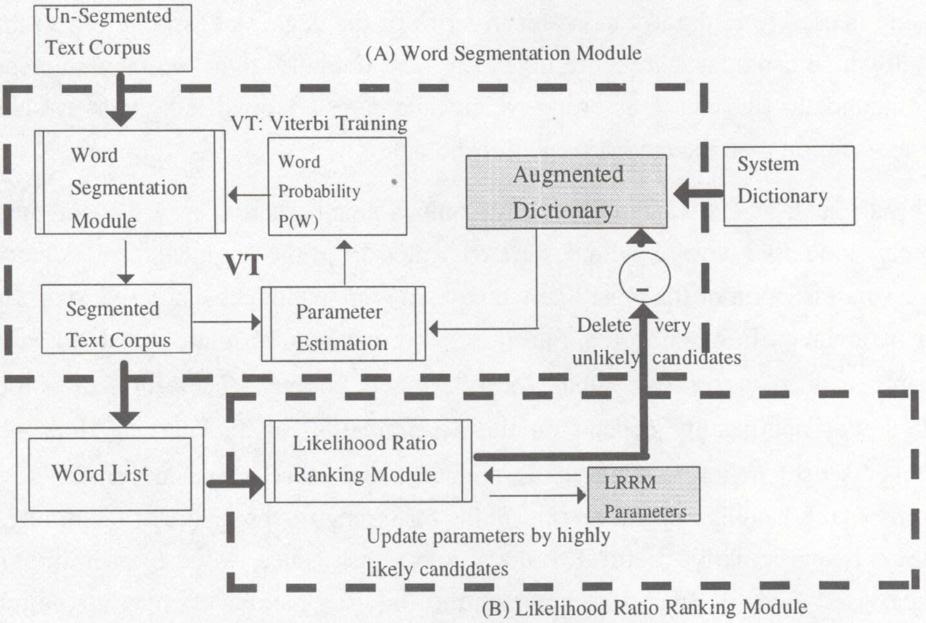


Figure 1 *Block diagram of the Chinese new lexicon identification system.*

Figure 1 illustrates the block diagram of the Chinese new lexicon identification system proposed in this paper. Dashed box A corresponds to the word segmentation module, and the 'filter' (i.e., the Likelihood Ratio Ranking Module) is represented by dashed box B. The other blocks not in the dashed boxes are either part of input or output of the system.

Initially, an *augmented dictionary* is formed by combining the system dictionary and all the n-grams that occur at least 5 times in the un-segmented input text corpus; such n-grams are the initial guesses of the 'potential unknown words', and their word probabilities are estimated as their relative frequencies in the corpus. The initial probabilities are then used to segment the input corpus. Afterward, the word probabilities are re-estimated from the best segmentation patterns, which are explored using the augmented dictionary. The re-estimation process, called Viterbi Training [Rabiner 93], is then repeated until the segmentation patterns no longer change or a specified number of segmentation iterations is reached.

The generated word list is then fed into the Likelihood Ratio Ranking Module (LRRM). A fraction of the word list entries which are judged very unlikely to be words is truncated from the augmented dictionary. (The '-' sign in the circle means to truncate some very unlikely n-grams, ranked by the Likelihood Ratio Ranking Module, from the augmented dictionary.) In addition, a fraction of the word list entries which are judged very likely to be words is used to update the parameter of the ranking module itself. Given the refined augmented dictionary, the Viterbi Training process is re-applied to find a better segmentation pattern and, thus, a better output word list. The process is then repeated so that the two modules can iteratively enhance one another's discrimination power by progressively refining the augmented dictionary and the parameters of the ranking module, and thus improve the system performance. Before investigating the individual effects of the segmentation module and the ranking module, the performance evaluation criteria and the method of evaluation are outlined in the following sections.

2.2 Performance Evaluation Criteria

In an unknown word identification task, it is desirable to recover from the corpus as many real new words as possible; in addition, the extracted word list should contain as few spurious word candidates as possible. The ability to extract all the new real words in the corpus is evaluated using the recall rate; on the other hand, the ability to exclude spurious words from the extracted word list is defined in terms of the precision rate. The new word precision rate, p, and the new word recall rate, r, are defined as follows:

$$p = \frac{\text{number of reported new words in the output list that are truly new words}}{\text{number of reported new words in the output list}}$$

$$r = \frac{\text{number of reported new words in the output list that are truly new words}}{\text{number of truly new words in the corpus}}$$

The precision and recall rates are, in many cases, two contradictory performance indices, especially in simple filtering approaches. When one performance index is raised, another index might be degraded. To make a fair comparison, the *weighted precision-recall* (WPR), which reflects the average of these two indices, is proposed here to evaluate the joint performance of precision and recall:

$$WPR\left(w_p : w_r\right) = w_p * \mathrm{p} + w_r * \mathrm{r} \qquad (w_p + w_r = 1),$$

where w_p, w_r are weighting factors for precision and recall, respectively. The F-measure (FM) [Appelt 93, Hirschman 95, Hobbs 96], defined as follows, is another joint performance metric which allows lexicographers to weigh precision and recall differently:

$$FM(\beta) = \frac{\left(\beta^2 + 1\right) p\, r}{\beta^2 p + r},$$

where β encodes user preference for precision or recall. When β is close to 0 (i.e., FM is close to p), the lexicographer prefers the system with higher precision; when β is large, the lexicographer prefers the system with higher recall. We will use Wp=Wr=0.5 and β=1 throughout this work, which means that no particular preference over precision or recall is imposed. If β=1, FM reduces to $FM = \frac{2pr}{p+r}$, which appreciates the balance between precision and recall in the sense that equal precision and recall is most preferred when $p + r$ is kept constant.

2.3 Experiment Environments and Evaluation Method

In our experiments, the un-segmented Chinese text corpus contains 311,591 sentences (about 1,670,000 words), which come from the China Times Daily News. Since most Chinese words are less than 4 characters long, only bigrams, trigrams and quadragrams (i.e., words of 2, 3 and 4 characters) are considered as word candidates. Furthermore, only an n-gram whose frequency is equal to or greater than 5 is considered as a word candidate since n-grams that rarely occur are considered to be less useful even though they are identified by the system. There are 242,042 distinct n-grams whose frequency of occurrence is equal to or greater than 5 in this corpus, including 99,407 bigrams, 99,211 trigrams and 43,424 quadragrams.

The system dictionary is a combination of the Academia Sinica dictionary [CKIP 90] and the BDC electronic dictionary [BDC 93] (excluding words which never appear in the un-segmented text corpus because such words will never be involved in the current task.) The merged dictionary contains 22,742 bigram words, 5,403 trigram words and 4,568 quadragram words, which add up to 32,713 entries.

It is difficult to know exactly how many new words can be found in a large text corpus unless the corpus is manually inspected. Therefore, most lexicon extraction researches [Chiang 92, Lin 93b, Tung 94, Wang 95] have not reported the recall rate. To estimate the precision and recall rates, a sample corpus of 1,000 sentences was randomly sampled from the input text corpus. These sentences were segmented manually, by a linguistics department staff of the Behavior Design Corporation, so that we could tell which n-grams in these samples could be considered words and which n-grams should be labelled as non-words. We then use the segmentation results generated from the sampled sentences to estimate the precision and recall of the system.

Although it is possible for different people to have different segmentation preferences (as demonstrated in [Sproat 96]), various approaches in this paper, however, are compared against the same segmentation benchmark prepared by the same person. Therefore, the relative differences in performance among the various approaches will have small statistical variations and are very likely to reflect the true situation. (Hypothesis testing, as given in the Appendix, was conducted to ensure that the algorithmic improvement of the proposed method is statistically significant. To estimate the various performance indices more precisely, though, we are planning to obtain a larger manually segmented corpus, such as the Academia Sinica Balanced Corpus [Huang 95], for evaluation of the system performance in our future work.)

This sample corpus contains 44,560 words. The numbers of distinct n-grams for n= 2, 3, 4 are 8,730, 9,658 and 8,745, respectively. Among these n-grams, 2,306 bigrams, 582 trigrams, and 394 quadragrams are recognized as distinct words in the manually segmented sample sentences. Among these words, 971 bigrams, 424 trigrams and 331 quadragrams are not registered in the system dictionary and are, thus, considered as the new words in the sample corpus. However, since only those n-grams that occur more than 5 times in the entire input corpus are regarded as word candidates in the current task, only these candidates will be extracted by the system; therefore, the performance is estimated against such n-grams. In other words, by 'new words', we are actually referring to 'new words that occurs at least 5 times in the input corpus'. Such 'new words' in the 1,000 sampled sentences include 866 bigrams, 295 trigrams, and 275 quadragrams, respectively; and these numbers are used as the 'number of truly new words in the corpus' to estimate new word precision/recall.

3. Segmentation Model for New Lexicon Identification

Given an input Chinese text corpus, the (new) words in the text can be extracted by segmenting the input text into word segments first, hoping that they are all correct, and we can then construct the (new) word list from the set of segments in the text corpus. 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 in the literature. Considering the capability of automatic training, adaptation to different domains, systematic improvement, and the cost for maintenance, probabilistic approaches are more attractive for large-scale systems. Furthermore, probabilistic segmentation models have been reported to be quite satisfactory [Chang 91, Chiang 92] when there are no unknown words in the corpus. Therefore, a probabilistic approach is adopted in this module.

3.1 The Statistical Word Segmentation Model

Given a string of Chinese characters $c_1, c_2, \ldots, c_n$, represented as c_1^n, the best word segmentation pattern S^* (based on the *vocabulary* V), is defined as the segmentation pattern which has the maximal likelihood value among all possible segmentation patterns S_j:

$$S^*(V) = \underset{S_j}{\operatorname{argmax}}\, P\left(S_j = w_{j,1}^{j,m_j} | c_1^n, V\right),$$

where $w_{j,1}^{j,m_j} \equiv \left\{ w_{j,1}, w_{j,2}, \ldots, w_{j,m_j} \right\}$ is the concatenation of the m_j words in the j-th alternative segmentation pattern S_j, and V is the *vocabulary* of the system used in the segmentation process (i.e., the set of words used to explore various segmentation patterns). To make estimation easier, the likelihood function is simplified as [Chang 91]:

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

which, in spite of its simplicity, has been shown to be effective [Chiang 92] in comparison with other rule-based or statistical models. The *vocabulary* is explicitly included in the model to highlight the fact that the `best' acquirable segmentation for an input corpus is a function of the vocabulary, which is used to explore the set of all possible segmentation patterns. Currently, our vocabulary corresponds to the set of n-grams in the augmented dictionary.

In the unsupervised mode of operation, the probabilities $P(w_{j,i} \mid V)$ are not known in advance and must be estimated from the input corpus. We will use the maximum likelihood (ML) criteria as described in the next section for this purpose. In the ML estimation process, the likelihood value that the corpus consists of the selected sequence of word segments is maximal among all the possible sequences. Since a word must be combined with other words to form sentences, whether an n-gram will be identified as a word depends on its context; therefore, the segmentation process virtually imposes contextual constraints on the possible word sequences.

3.2 Viterbi Training for Parameter Re-estimation

The word probabilities in Equation (1) can be estimated from a large pre-segmented corpus if one is available. However, such a training corpus is usually too expensive to construct. Therefore, an unsupervised training method, called Viterbi Training (VT) [Rabiner 93], is adopted in the current work to estimate the probabilities.

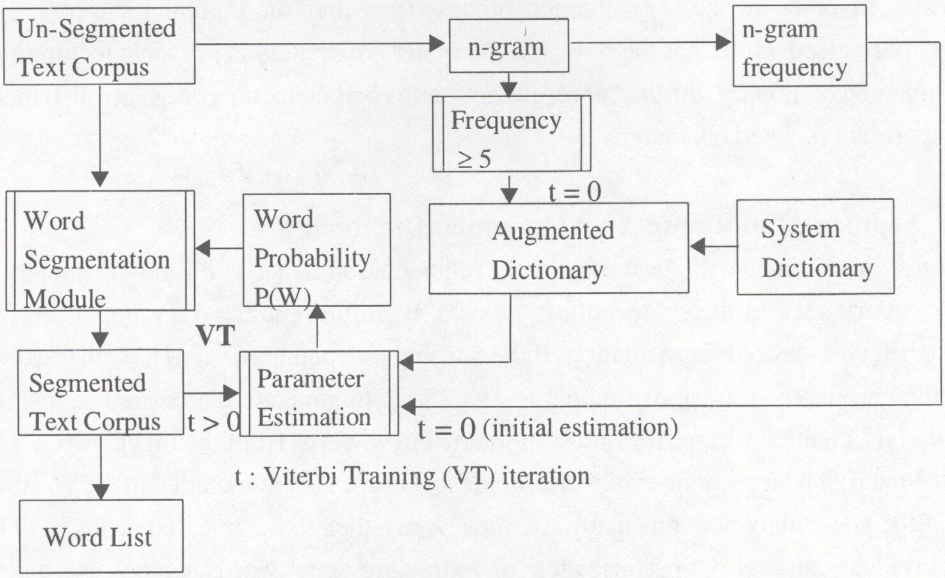


Figure 2 *Viterbi Training model for word segmentation using an augmented dictionary.*

Figure 2 shows the block diagram of the word segmentation module, which applies the Viterbi training process to achieve the maximum likelihood segmentation of the corpus. The initial set of n-grams which occur at least 5 times in the input corpus is acquired from the un-segmented text corpus with their frequencies. They are combined with the system dictionary words to form the initial augmented dictionary (i.e., the word candidates used to segment the corpus.) The initial probabilities of all the n-grams in the augmented dictionary are estimated as the relative frequencies of the n-grams in the input text corpus. Various segmentation patterns of the corpus are then explored in terms of the known words and unknown word candidates in the augmented dictionary. The path (i.e., the segmentation pattern) with the highest likelihood value is marked as the best path for the current iteration. A new set of parameters is then re-estimated based on the best path obtained by the Maximum Likelihood (ML) Estimator. This process repeats until the segmentation patterns no longer change or a maximum number of iterations is reached. We can then derive the word list from the segmented text corpus.

As indicated in Figure 2, only n-grams which occur at least 5 times in the corpus will be included in the augmented dictionary; this restriction is intended to remove n-grams that are rarely used even though they might be judged as words. The reasons for applying this restriction are as follows. First, the statistics of such n-grams are usually too unreliable for use. Second, there are a lot of n-grams which occur only once or twice, most of which are insignificant. By removing such n-grams, the number of word candidates involved in the segmentation process (and thus the computation cost) can be greatly reduced. Note that the words in the system dictionary are always included in the augmented dictionary. Furthermore, all the single characters (unigrams) are also included to represent isolated characters.

3.3 Segmentation Using An Augmented Dictionary

As described earlier, the best acquirable segmentation depends on the vocabulary. The vocabulary used in the segmentation process, therefore, plays a very important role in acquiring the desired segmentation. If the vocabulary contains *all and only* the real words in the corpus, then the above model is very likely to generate the desired segmentation and, thus, enable us to retrieve most of the real new words from the input corpus. On the other hand, if a large number of words in the text corpus is not included in the vocabulary, or if the vocabulary contains many spurious words, then the segmented results will be far from ideal, and good performance in extracting new words cannot be expected. Therefore, if we are able to make good guesses on the real vocabulary progressively, then the unknown words will probably be extracted with progressively higher recall and precision rates.

However, most segmentation models (such as [Chang 91]) are based on the assumption that there are no unknown words in the text corpus. Hence, the unknown words are usually broken down into combinations of short segments (including single characters) after the segmentation process. Under such circumstances, an intuitive way of identifying new words is to merge adjacent segments into longer n-grams [Tung 94, Wang 95], then to use a separate filter to filter out inappropriate candidates, and, last, to optionally apply a disambiguation step to remove candidates that overlap other candidates in the sentences [Tung 94].

There are some problems with such a non-iterative segmentation-and-merging process, which uses known words alone to find unknown word candidates. First, there are two types of segmentation errors [Chiang 92, Lin 93a, 93b], namely over-segmentation (which segments unknown words into shorter segments) and over-merging (which combines short words into longer ones). The merging operation may not recover all the mis-segmented words because it can only recover unknown words that are over-segmented into small segments. Such an operation, however, cannot recover unknown words that are (fully or partially) embedded in known words. For instance, the following example in Section 1.1: ' 土地 公有 政策 ...' ('The policy for the land to be owned by the public...') is likely to be segmented into ' 土地公 有 政策 ' ('The Local Village God has his own policy...') if we don't have the unknown word ' 公有 ' ('public ownership') in our vocabulary and if, instead, we have a known word ' 土地公 ' ('the Local Village God') which includes the first character of the unknown word ' 公有 '. In this case, merging of the adjacent segments ' 土地公 ', ' 有 ' and ' 政策 ', will never produce the unknown word candidate ' 公有 '. Second, a large number of merged segments will be produced, which probably will not be produced if a reasonably good augmented dictionary is used during the segmentation process. Submitting such a large candidate list (which amounts to about 2 million entries for our corpus) to the filters will degrade the precision of the system. Finally, an extra disambiguation process, as described in Section 1.1, might be necessary to resolve ambiguities between overlapping candidates.

To avoid the above problems, we can construct an *augmented dictionary* which includes potential unknown words to the vocabulary in addition to known words, and make the known words and potential unknown words compete with each other under the contextual constraints imposed by the segmenter. It can then be expected that the number of words in the output list (derived from the segmented corpus) will be greatly reduced after applying the contextual constraints within the sentences because only 'likely' unknown words (in the maximum likelihood sense) among all the potential unknown words (in the augmented dictionary) will be submitted to the filter. With our corpus, the number of words derived in the initial iteration using the augmented dictionary is only about 1/15 that of the merged word candidates (which are acquired by segmenting the corpus using known words only and merging adjacent n-grams afterwards.)

To focus on the extraction of significant new words that occur frequently, the initial *augmented dictionary* used for segmentation consists of single characters in the corpus, the words in the system dictionary, and *n-grams* whose frequency of occurrence is at least 5 in the text corpus. The augmentation, thus, is able to recover not only over segmented unknown words, but also short words that are embedded in known words. Furthermore, only bigrams, trigrams and quadragrams are included in the augmented dictionary since most Chinese words are 2, 3, or 4 characters long.

Applying MLE (maximum likelihood estimation) to the segmentation patterns using the augmented dictionary, the most likely n-grams (in ML sense) will include not only the system dictionary words, but also the most likely unknown words. To view this in another way, use of the augmented dictionary will, in some sense, merge (or split) known words and isolated single characters into potential unknown word candidates. Such candidates are more likely to be words than are the millions of randomly merged n-grams because they are selected based on the ML criterion under the contextual constraints embedded in the input corpus. Consequently, only such very likely n-grams, instead of the millions of randomly merged n-grams, will be submitted to the filter for further justification; hence, the precision of the system is expected to improve significantly. Besides, it is also unnecessary to apply an extra disambiguation step to remove candidates that overlap other candidates in the sentences ([Tung 94]) since the segmentation processes will automatically determine which n-grams will survive based on the ML criterion and, thus, will resolve such ambiguity implicitly.

3.4 Performance Evaluation on Viterbi Word Segmentation

To show how effective the unsupervised Viterbi word segmentation module is, the performance of only using the segmentation module is listed in Table 1. It is observed that the training process converges very quickly using the above Viterbi training procedure. (See Figure 3 at the end of this section for the bigram example.) The performance of the initial iteration (iteration #1), the second iteration and the 13th iteration (where the system converges) is listed in the table.

n-gram	iteration number	p (%)	r (%)	WPR	FM
2	1	65.83	72.75	69.29	69.12
	2	68.67	76.67	72.67	72.45
	13*	68.72	78.41	73.57	73.25
3	1	26.45	78.64	52.55	39.59
	2	28.81	80.68	54.75	42.46
	13*	29.63	81.36	55.50	43.44
4	1	36.57	93.09	64.83	52.51
	2	38.24	93.45	65.85	54.27
	13*	38.96	93.09	66.03	54.93

Table 1. *Performance of the word segmentation module using the Viterbi training procedure for new word identification (*: performance at converge; convergence is reached at iteration #13)*

Table 1 shows that the precision rates, after convergence for the extracted new words which are 2, 3, and 4 characters long, are about 69%, 30%, and 39%, respectively. The recall rates for the new words are about 78%, 81%, and 93%, respectively. The joint performance in WPR is about 74%, 56% and 66%, and FM achieves rates of 73%, 43% and 55%, respectively, for 2-, 3-, 4-character new words. This means that most (78-93%) new words are included in the extracted lists, and that one can pick up a correct new word from the lists about once every 1.5 - 3.4 entries. This unsupervised segmentation-only model is, therefore, a useful tool by itself for extracting new words from a corpus.

To give the readers some feeling whether this performance is good, it was compared with other previous works. Since most other works we have investigated either do not provide precision and recall performance at all, or simply provide an estimate of the precision rate without giving an estimate of the recall rate (which is usually costly to estimate), we can therefore only quote the results reported in [Wang 95] for a very rough comparison. (The criteria used by different lexicographers in recognizing an n-gram as a lexical entry may be different. Therefore, the following comparison may not be a fair comparison since it is not based on completely identical environments.) In [Wang 95], a measure similar to the *strength* statistic used in the Xtract system [Smadja 93] was used to extract new words. The corpus has the same domain as ours (i.e., news articles) and has about the same size. The reported new word precision rates are about 22%, 28% and 12%

for bigram. trigram and quadragram new words. (The new word recall rate was not reported in the above mentioned work. Note also that the terms 'bigram', 'trigram' and 'quadragram' in [Wang 95] are not completely identical to '2-character', '3-character' and '4-character' strings, respectively. But the distributional analyses and examples given in [Wang 95], such as 'trigram proper names' and 'bi-collocation', seem to suggest that, most of the time, they are equivalent. Even with such variation, it is expected that the precision for bigrams, trigrams and quadragrams in the above work should not exceed 30%.) With the high recall rates and much higher precision rates, the proposed Viterbi training approach should be competitive. We will address later how such performance can be improved even further. (In the improved scheme, it is possible to increase the precision rates to about 72% (bigram), 39% (trigram) and 56% (quadragram) after 21 iterations.)

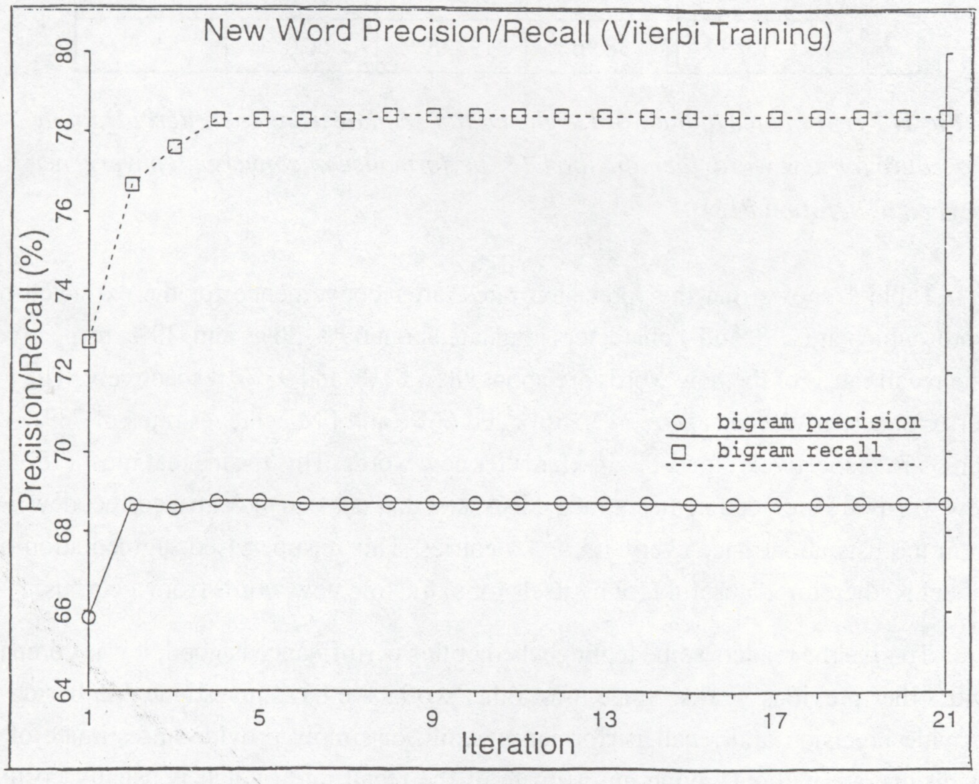


Figure 3 Precision and recall rates for bigram new word identification in each iteration.

To show how the Viterbi Training converges, the performance of the bigram new words for each iteration is shown in Figure 3. It shows that both the precision and recall of the bigrams are improved as Viterbi training progresses, instead of sacrificing precision for recall or *vice versa*, as is often observed in other simple filtering approaches.

Note that the largest changes in precision and recall occur between the first and the second iterations. No further significant change is observed in the performance curves. This means that two iterations are usualiy enough for Viterbi Training. This situation is also shown in Table 1, in which the differences between the second iteration and the converged results is negligible.

4. Using Association Metrics for Refining the Vocabulary

4.1 Problems with Using the Segmentation Module alone

In spite of the encouraging results described in the last section, the above model has one main problem: the likelihood function used in the maximization process consults only n-gram frequency information. No other character association features (and syntactic or semantic information) are consulted during the word segmentation process. Therefore, the 'most likely' segmentation thus acquired is simply the one which has the largest product of relative word frequencies among all the segmentation patterns. However, the 'new words' extracted in this way may not have the desirable characteristics, such as 'having high mutual information', 'containing no special characters (such as 'de' (的), 'zhi' (之)), symbols, and Chinese quantifiers', and so on.

In other words, the words extracted from the 'most likely' segmentation pattern based on the above likelihood function may not coincide with lexicographers' general sense of *most likely words*. For instance, the bigram '的人' ('people who ...'), which occurs 962 times in the corpus, may be recognized by the word segmentation module as a word simply because it occurs frequently. As a result, many n-grams which will not be qualified as words by lexicographers may be extracted in the word segmentation process simply because they occur frequently, and the precision rate will be low under the lexicographers' criteria. (For more examples and quantitative analyses, please refer to Section 6.2, "Error Analyses".) Therefore, it is desirable to use other information, such as some association measures, to remove such n-grams from the augmented dictionary.

4.2 Truncating Inappropriate New Word Candidates: A Two-Class Classification Approach

One way to attack the above-mentioned problem is to filter out non-word candidates using a filter by means of some association metrics. In this filtering approach, only n-grams which have high association above a pre-defined threshold will be recognized as a word. However, in traditional works of this kind, the association metrics, such as mutual information and entropy, are usually independently applied in several filtering stages without jointly considering them. It is known in information theory, however, that joint consideration of multiple features provides more discriminative information for classifying n-grams [Chiang 96]. For instance, if we first use frequency information to filter out low frequency candidates and then use mutual information to filter out low mutual information n-grams in the remaining candidates, we may end up truncating words which have high association but are less frequently observed. In addition, the thresholds for the filters are usually determined heuristically. Hence, they may not be applicable to other domains. Therefore, a minimum error classifier, which can jointly integrate various association metrics into one *joint* metric, is implemented in the current work as our 'filter' to supplement the Viterbi word segmentation module.

4.2.1 Filter Design: A Likelihood Ratio Ranking Model for Joining Association Metrics

To filter out unlikely candidates, a likelihood ratio ranking model is used to measure whether a character n-grams is more likely to be a word than a non-word. To identify whether an n-gram belongs to the word class (W) or the non-word class ($\overline{\text{W}}$), each n-gram is associated with a feature vector **x** which is derived from the statistics of the n-grams in the un-segmented corpus. It is then judged to determine whether it is more likely to be generated from a word model or a non-word model using the following likelihood ratio test:

$$\lambda = \frac{f(\mathbf{x} \mid \mathbf{W})P(\mathbf{W})}{f(\mathbf{x} \mid \overline{\mathbf{W}})P(\overline{\mathbf{W}})}, \qquad \text{LLR} \equiv \log \lambda \ ,$$

where λ is the likelihood ratio for the n-gram, $\log \lambda$ is the log-likelihood ratio (LLR), $f(\mathbf{x} \mid \mathbf{W})$ (or $f(\mathbf{x}|\overline{\mathbf{W}})$) is the density function of the feature vector **x** in the word-class (or non-word class), and $P(\mathbf{W})$ (or $P(\overline{\mathbf{W}})$) is the prior probability for the corresponding class **W** (or $\overline{\mathbf{W}}$). The numerator in the above ratio is the likelihood that **x** will be generated from the word-class, and the denominator is the likelihood that it will be drawn from the non-word class. If $\lambda \geq 1$, i.e., the log-likelihood ratio LLR ≥ 0 (or larger than a threshold $\log \lambda_0$) for an n-gram, then it is classified as a word; otherwise, it is classified as a non-word. Such a likelihood ratio test is known to be `the most powerful test' [Papoulis 90] in testing two hypotheses; it is also known to be the minimum error classifier [Devijver 82, Juang 92] in pattern recognition. Hence, we can use the *log-likelihood ratio* as a *joint* association metric of the other commonly used association metrics and use this joint metric to determine which n-grams are more likely to be words and which are more unlikely to be words among all n-grams. Therefore, we use such a model as our basis for the *likelihood ratio ranking module* (LRRM, or 'ranking module' for short).

To estimate the density functions and the prior probabilities, we must have two sets of training n-grams, in one of which all the class members are assigned the word-class label, and the other set has class members that are labeled as non-word. However, since the n-grams in the input text corpus are not associated with their word/non-word class labels, the initial class labels of the feature vectors are obtained by dividing the n-grams of the input text corpus into word and non-word classes, depending on whether an n-gram can be found in the system dictionary or not.

4.2.2 Character Association Features for Evaluating Likelihood Ratio

To estimate the LLR's (log-likelihood ratios) of the character n-grams, many discriminative features, such as mutual information [Church 90, Wu 93, Su 94], entropy [Tung 94], dice metric [Smadja 96], relative strength [Smadja 93, Wang 95] and χ^2 testing statistics [Papoulis 90], can be used in the ranking module. In particular, the mutual information and entropy features have been found to be useful for English compound word extraction [Chang 97]. Therefore, they are used in the current work. The definitions of the two association metrics are given as follows.

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

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

where P(x) and P(y) are the prior probabilities of the individual characters x and y, and P(x,y) is the joint probability that the two characters will co-occur in the same bigram. The numerator in the formula is the probability that the characters will appear jointly, and the denominator is the probability that the individual characters will occur independently. This measure is, therefore, an indicator of how likely it is that individual characters will occur jointly comparing to the cases where they co-occur incidentally. If the mutual information measure is much larger than 0, then the bigram tends to have strong association. To deal with n-grams with n greater than 2, the idea of dependent *vs*. independent probabilities was extended to the following definition for trigram mutual information [Wu 93, Su 94]:

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

$$P_I = P(x)P(y)P(z) + P(x)P(y,z) + P(x,y)P(z) \ .$$

In the above definition, the numerator P_D denotes the probability that three characters will be jointly dependent (i.e., the probability that three characters will form a 3-character word), and the denominator P_I denotes the total probability that three characters will belong to different words. The extension can be made to other n-grams in a similar way.

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

$$H_L(x) = -\sum_{c_i} P_L\left(c_i;x\right) log P_L\left(c_i;x\right)$$

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

where $P_L(c_i;x)$ and $P_R(x;c_i)$ are the conditional probabilities of the left (L) and right (R) neighboring characters of the n-gram x, respectively (and c_i refers to the left/right neighboring characters). There are several different ways to combine the left and right entropies for the classification task. In this paper, we will not focus on the use of any particular features; therefore, the average of the left and right entropies, denoted by H (i.e., $H(x) = \left(H_L(x) + H_R(x)\right) / 2$), is used as a joint feature.

The joint density functions $f(\mathbf{x} \mid \mathbf{W})$ and $f(\mathbf{x} \mid \bar{\mathbf{W}})$ in the log-likelihood ratio are modelled as a mixture of multivariate Gaussian distributions. In other words, we have [Roussas 73]:

$$f(\mathbf{x} \mid \mathbf{W}) \equiv \sum_{i=1}^{M} r_i \cdot N\left(\mathbf{x};\mu_i,\Sigma_i\right), \quad \sum_{i=1}^{M} r_i = 1, \ \mathbf{x} = [\mathbf{I}, \mathbf{H}]$$

$$N(\mathbf{x};\mu,\Sigma) = (2\pi)^{-D/2} \left|\Sigma\right|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^{T}\Sigma^{-1}(\mathbf{x}-\mu)\right],$$

where $f(\mathbf{x} \mid \mathbf{W})$ is the density function of the feature vector $\mathbf{x}$ for the word-class, **I** is the mutual information , **H** is the average entropy. M is the number of mixtures, r_i is the prior probability for the *i*-th mixture, $N\left(\mathbf{x};\mu_i,\Sigma_i\right)$ is the multivariate Gaussian density function for the *i*-th mixture, μ_i is the mean vector, Σ_i is the covariance matrix for the *i*-th mixture, and D is the number of features used (i.e., the dimension of features) in the feature vector $\mathbf{x}$ (currently, D=2). The prior probabilities, mean vectors, and covariance matrices (i.e., the *parameters*) for the two classes can be estimated using an unsupervised method [Duda 73], which will not be addressed here. By using such a density function, the features will be jointly considered for decision making, and the correlation between the two features is taken into account in terms of the covariance matrices. In the current work, 3 Gaussian mixtures (M=3) are used for the multivariate Gaussian density function of each class. Of course, if the density functions are not multivariate Gaussian mixtures, other families of density function may have to be used [Chang 97]. Such general parameter estimation issues, however, will not be addressed here.

4.2.3 Ranking-Module-Only Performance

The likelihood ratio ranking module (LRRM) can be used alone to identify n-grams as words or non-words. Therefore, it is interesting to see how this module performs by itself, before combining it with the segmentation module. One intuitive way to apply the ranking module is to use it in the unsupervised mode of operation as follows. Initially, the system dictionary is used to assign the word and non-word labels to all n-grams. The parameters for the two classes are then estimated according to such class labels. We then use $\log \lambda = 0$ as the threshold to identify the n-grams as word or non-word, and use the newly identified class labels to re-estimate the parameter values. Table 2 shows the following estimated performance (against n-grams in the 1,000 sample sentences described in Section 2.3) after 21 iterations.

n-gram	p (%)	r (%)	WPR	FM
2	54.28	90.99	72.63	68.00
3	33.78	63.01	48.40	43.98
4	51.17	81.42	66.30	62.84

Table 2. *Performance of the likelihood ratio ranking module (two-class classifier) (with* $\log \lambda = 0$ *as the threshold) for identifying unknown words.*

In comparison with the results shown in Table 1, where only Viterbi training is applied to word segmentation, the bigram WPR using the ranking module alone is slightly worse, and FM is worse by 5%. For trigrams, the slightly better precision is offset by the large loss in recall; therefore, WPR is smaller by 7%, and only 0.5% is gained in FM. For quadragrams, the precision is higher (by 12%), but the recall is worse (by 12%), which results in a slightly larger WPR but a larger difference (8%) in FM. It is hard to determine whether using the ranking module alone is better than using only the word segmentation module from such differences especially when we are not sure whether the threshold is the best for the various performance indices. It depends on whether we are comparing WPR or FM and whether we are comparing bigrams, trigrams or quadragrams (although the two models are both better than a few other reported works.) However, the following reasons suggest that using the ranking module alone is not a good idea under the current unsupervised mode of training.

First, the parameters of the ranking module may not be reliably estimated without a large training corpus with correctly assigned word class labels. Second, the ranking module by itself does not take advantage of the contextual constraints in identifying new words; in other words, each n-gram is identified independently of its neighboring n-grams. Therefore, some mis-classified instances, such as the '委會同' example in Section 1.1, can be avoided if the contextual constraints are taken into consideration. Finally, the LLR=0 threshold may not be the best threshold for achieving the highest precision, recall, WPR or FM [Chang 97] (although it normally provides a good starting point, in the minimum error sense, for training the system parameters toward the best precision/recall performance.) Therefore, the ranking module will not be used as a stand alone classifier in the current work. Instead, it will be used simply as a ranking device, as described in Section 2, to supplement the segmentation module in acquiring better segmentation patterns and better word lists.

4.3 Combining Word Segmentation Module and Ranking Module

One way to make the ranking module benefit from the contextual constraints provided by the word segmentation module is to cascade the word segmentation module and the ranking module together, and use the ranking module to truncate unlikely new words from the output of the segmentation results (not from the augmented dictionary as we will do in the method proposed in Section 5.) Table 3 shows the performance obtained by applying Viterbi training twice to get the segmentation output and then truncating the worst 10% (in terms of their LLR ranks) of the new words extracted from the segmentation output. Viterbi training is applied only twice since further iterations would not significantly change the segmentation performance (as mentioned in Section 3.4). Furthermore, instead of using an absolute threshold for the log-likelihood ratio test, we use a *relative mode* of filtering to truncate only the most unlikely 10% new words from the output of the segmentation since the best thresholds (in terms of maximum WPR or FM [Chang 97]) cannot be reliably estimated in unsupervised training. To see the effects, the performance of the segmentation-only (WS-only) model (Table 1) and the ranking-module-only (LRRM-only) model (Table 2) are re-cited here for comparison. The performance of this cascading scheme is entitled 'Non-Iterative' in the table since the word segmentation module or the ranking module is applied only once. The shaded cells in Table 3 designate the best WPR or F-measure performance of the various models.

n-gram	Models	p (%)	r (%)	WPR	FM
2	WS-only	68.72	78.41	73.57	73.25
	LRRM-only	54.28	90.99	72.63	68.00
	Non-Iterative	73.56	73.90	73.73	73.73
3	WS-only	29.63	81.36	55.50	43.44
	LRRM-only	33.78	63.01	48.40	43.98
	Non-Iterative	31.90	80.34	56.12	45.66
4	WS-only	38.96	93.09	66.03	54.93
	LRRM-only	51.17	81.42	66.30	62.84
	Non-Iterative	42.38	93.09	67.74	58.25

***Table 3.** Performance obtained by cascading the segmentation module and the ranking module, in comparison with other models (WS-only: segmentation only; LRRM-only: likelihood ratio ranking module only; Non-Iterative: cascading the two modules and truncating the most unlikely 10% new words from the segmentation output). (Note: the WS model applies 13 iterations to converge, and LRRM model is iterated 21 times.)*

From the above table, the performance obtained by cascading the two modules is better than that of the segmentation-only model (which converges at the 13th iteration) and the ranking-module-only model (which iterates 21 times) in terms of WPR and FM, with only one exception. There are some implications from the above observation. First, the cascading scheme does have some degree of integration between the segmentation module and the ranking module, as can be expected in most such cascading schemes. In comparison with Table 1, it is easy to see that the improvement is gained by sacrificing a little recall for higher precision. Second, we can truncate unlikely candidates using *relative mode of filtering* without really depending on an optimal threshold (in whatever sense) under the current unsupervised mode of training. For this reason, the classifier actually operates as a likelihood ratio ranking module in our system, which works in relative mode of filtering. Note that the recall rates, which might drop slightly in the filtering step, will be compensated by using an iterative scheme, as described in the following sections, so that both precision and recall can be improved at the same time.

5. An Iterative Approach to Integrating the Word Segmentation Module and the Likelihood Ratio Ranking Module

5.1 Problems with Non-iterative Approaches

Although the above cascading scheme for combining the word segmentation module and the filter (ranking module) is easy to implement, it is non-iterative in the sense that there is no feedback path to the previous stage. As a result, the information provided by the filter cannot be used to enhance the power of the segmentation module. On the other hand, the parameters of the filter are estimated independently of the segmentation results; hence, the segmentation output has no contribution to the performance of the filter. The information provided by one of the two modules, thus, cannot fully supplement or enhance the other.

To address this problem, note that the performance of the word segmentation module greatly depends on how well potential unknown words are included to the augmented dictionary; and the performance of the filter greatly depends on how well the parameters for the filter are estimated. Ideally, we should use a dictionary consisting only of the words embedded in the input corpus to get the desired segmentation. However, it is impossible to create such a dictionary in advance. Therefore, the initial augmented dictionary is constructed simply by combining the system dictionary and the n-grams that occur at least 5 times in the input corpus. In other words, the association information for each entry in the augmented dictionary is not consulted in constructing the augmented dictionary. On the other hand, if the true class labels of the n-grams in the input corpus are known in advance, then the estimated filter parameters will be close to the desired values. However, we can only rely on the system dictionary to assign word-class labels with certainty; the other n-grams not in the system dictionary may be incorrectly labelled. As a result, the parameters of the filter are estimated independently of the segmentation results.

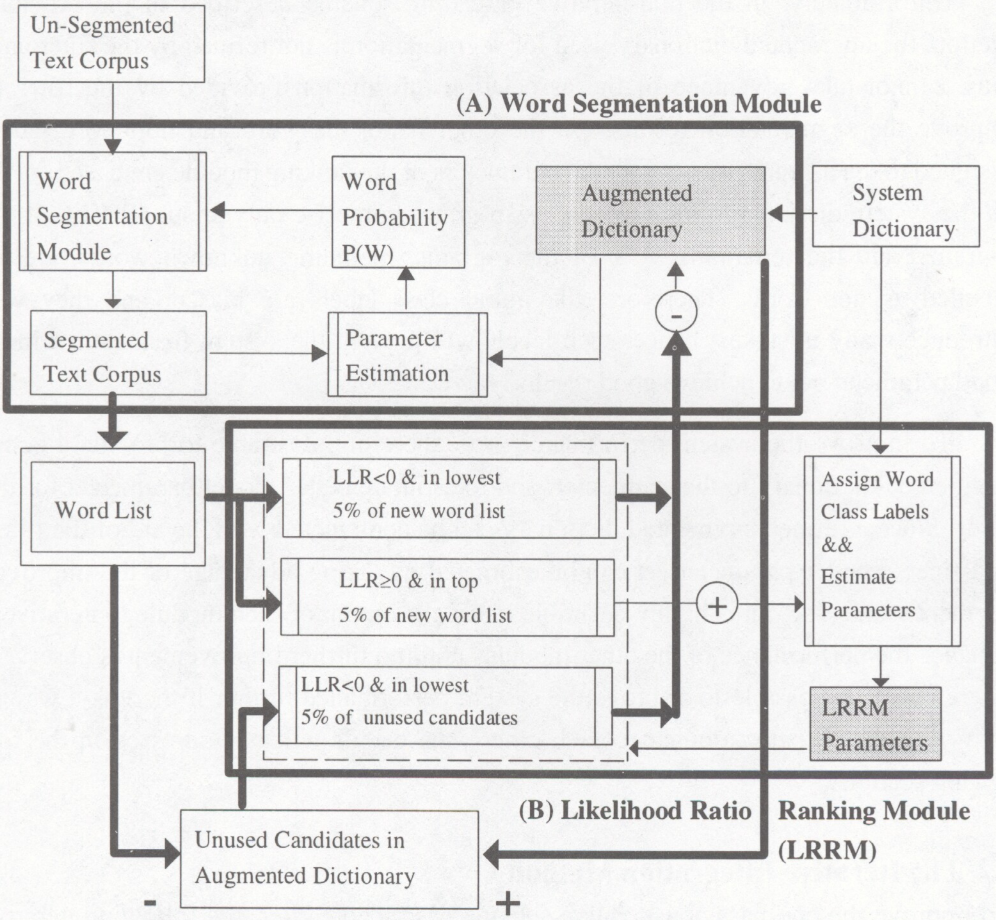
Therefore, it is desirable to use the filter to improve the performance of the segmentation module by refining the augmented dictionary. It is also desirable to refine the class labels of the n-grams in the input corpus (and, thus, the estimated parameters of the filter) by using the new word list suggested by the segmentation module.

Unfortunately, in the non-iterative cascading scheme described in the previous section, the augmented dictionary used for segmentation is not refined by the filter and, thus, cannot take advantage of the association information provided by the filter to improve the segmentation results. On the other hand, the word and non-word labels assigned to the n-grams to estimate the parameters of the ranking module are based solely on the system dictionary. Such word-class n-grams comprise only about 10% of all the n-grams, and the remaining 90% of the n-grams, including unknown words, are all labelled as non-words. Therefore, the initial class labels are biased, and they will introduce many mistakes; hence, such labels will prevent the system from estimating a good parameter set to achieve good results.

To improve the system performance, it is, therefore, desirable to provide a better augmented dictionary to the segmenter, and to estimate better model parameters for the filter. Since a refined augmented dictionary can be constructed with the aid of the filter, and since a better parameter set can be estimated by taking advantage of the improved segmentation results, it is highly desirable to use the output of each module to iteratively enhance the performance of the other modules until no further improvement is observed. Therefore, it is possible to improve the system performance further if there is a way to form a feedback path to enhance each other. The details will be addressed in the following sections.

5.2 The Iterative Integration Method

To improve the one-pass non-iterative scheme described earlier, the following iterative approach is proposed to integrate the word segmentation module and the likelihood ratio ranking module (LRRM). In each iteration, the word segmentation result is improved by using a refined augmented dictionary, which was produced by truncating the most unlikely word candidates in the augmented dictionary of the last iteration with the aid of the ranking module. Such improvement obtained by refining the augmented dictionary is possible since statistical word segmentation can achieve very satisfactory results (over 99% word segmentation accuracy [Chiang 92, Lin 93b]) if all the unknown words are included in the system dictionary and no spurious words are included in the system dictionary. On the other hand, the parameters of the ranking module are improved at the end of each iteration by updating the word and non-word class labels of the n-grams; this is accomplished by re-labeling the class labels of a small fraction of the most likely words identified by the word segmentation module. The performance of the ranking module can, thus, be improved by using progressively improved word/non-word class labels derived from the word segmentation output using the contextual information of the input corpus.



Un-Segmented Text Corpus
(A) Word Segmentation Module
Word Segmentation Module
Word Probability P(W)
Augmented Dictionary
System Dictionary
-
Segmented Text Corpus
Parameter Estimation
Word List
LLR<0 & in lowest 5% of new word list
LLR≥0 & in top 5% of new word list
+
LLR<0 & in lowest 5% of unused candidates
Assign Word Class Labels && Estimate Parameters
LRRM Parameters
(B) Likelihood Ratio Ranking Module (LRRM)
Unused Candidates in Augmented Dictionary
-
+

Figure 4 *Block diagram for iteratively integrating the word segmentation module and the likelihood ratio ranking module for unknown word identification.*

Figure 4 illustrates the iteratively integrated system for unknown word identification. Box (A) is the segmentation module as described in previous sections. Box (B) corresponds to the likelihood ratio ranking module, which is used to estimate the log-likelihood ratios for various n-grams in order to update the augmented dictionary and to update the word/non-word labels of the n-grams in the input corpus for re-estimation of the parameters of the ranking module. The three processes in the dashed boxes within the ranking module represent three updating actions (to be described later) for updating the augmented dictionary of the segmentation module and the parameters of the ranking module.

As described earlier, the initial augmented dictionary is constructed by combining the system dictionary and n-grams in the input corpus which appear more than 5 times (inclusive); their initial word probabilities are estimated as their relative frequencies in the input corpus. Each such n-gram is initially labeled as a 'word' if it belongs to the system dictionary; otherwise, it is labeled as a 'non-word'. The initial parameters of the filter are then estimated based on the initial class labels. Once the initialization is complete, the input corpus is segmented, and its output word list is used to update the augmented dictionary and filter parameters; afterwards, the refined augmented dictionary and filter parameters are used iteratively for re-segmentation and re-ranking.

Note that the Viterbi training process for word segmentation converges very quickly, and the largest change in performance occurs between the first iteration and the second iteration; further iterations do not significantly change the best segmentation results (see Table 1 for details). We, therefore, apply Viterbi training only twice in the word segmentation process to reduce the processing time.

The ranking module takes an n-gram list as its input and estimates the log-likelihood ratio (LLR) of each n-gram in the list. The log-likelihood ratios are then used to rank the n-grams and indicate which one is 'more likely' or 'more unlikely' to be a word; a small fraction of the input list is then used for rejection or acceptance. More precisely, 3 sublists (corresponding to the three updating actions in Figure 4) are submitted to the ranking module for ranking (by their LLR's); two such sublists are then used to update the augmented dictionary, and one sublist is used to update the parameters of the ranking module as follows.

(1) The new word n-grams derived from the segmented corpus whose LLR is less than zero (LLR<0) are sorted by their LLR's; the worst 5% (in terms of LLR) of this sublist are truncated from the augmented dictionary to refine the augmented dictionary. This step is intended to remove the most unlikely candidates in the new word list from the augmented dictionary using the global character association metrics. Such candidates are preferred by the segmentation module but unqualified by the ranking module; therefore, they should be removed to prevent them from appearing in the word segmentation output again in later iterations, so that the segmentation results in later iterations will be improved.

For instance, the non-word substring '不足爲' is identified as a new word in the incorrectly segmented phrase '不足爲 大哥' ('is not qualified as the Big Brother') since it also appears frequently in words like '不足爲奇', '不足爲由', '不足爲信', '不足爲惜' or '不足爲慮' ('is not really strange', 'cannot be accepted as the appropriate reasons', 'cannot be accepted as evidence', 'is not pitiful' or 'is not worth worrying about')). However, the non-word string '不足爲' (with LLR= -7.73) will not be qualified by the ranking module; therefore, it will be removed by this updating action, resulting in the better segmentation '不足 爲 大哥'. (Also recall the '(移送)台中少年法庭審理' example in the Introduction section, which acquires the correct segmentation '(移送) 台中 少年法庭 審理' after two successive removals of the non-word strings '台中少年' and '法庭審理'.)

(2) The potential unknown words (in the augmented dictionary) which do not appear in the segmented corpus form a sublist of 'unused candidates'. Among such 'unused candidates' whose LLR<0, the worst 5% n-grams (in terms of LLR) are also excluded from the current augmented dictionary. Such entries are removed because they are neither preferred by the word segmentation module nor preferred by the classification modules. Hence, they are very unlikely to be new words. (The '+' and '-' signs near the box entitled 'unused candidates in augmented dictionary' mean to form the 'unused candidates' by including ('+') potential unknown words in the augmented dictionary and then excluding ('-') entries that appear in the output word list of the segmentation module.)

Most such n-grams are random n-gram strings, such as '元遠期' (which is a substring of '美元遠期外匯' 'US dollar long-term exchange rate'). Removing such entries will reduce the computation cost without introducing significant errors. Given the refined augmented dictionary updated according to the above two sublists, the Viterbi training will be applied again, in the next segmentation-filtering iteration, to segment the corpus using the refined augmented dictionary.

(3) A third sublist is used to facilitate estimation of the filter parameters as follows. The parameters of the classifier are trained by dividing n-grams in the input corpus into word and non-word classes and estimating the parameters from the distributions of these two classes. Initially, only those n-grams in the system dictionary are assigned the word-class label for training the word-class parameters, and the others are assigned the non-word label for training non-word-class parameters. To improve parameter estimation of the likelihood ratio ranking module, those n-grams which were assigned the word-class label in previous iterations are first excluded from the new word list (as described in (1)); a small fraction of the remaining new words, which have the highest LLR's, are then re-assigned to the word-class so that the parameters of the ranking module are re-estimated based on the new class labels. The number of re-assigned new word candidates is 5% of the n-grams which have already been assigned the word class label. In other words, the number of n-grams which are assigned the word-class label will

increase by 5% per iteration. The reason for setting this fraction according to the current word-class size is to ensure that the number of new word-class members will not grow so fast as to overwhelm the number of original word-class members (which contain only the system dictionary entries).

Since such n-grams are preferred by both the segmentation module and the ranking module as words, they are combined with the original system dictionary entries to train the word-class parameters. The remaining entries are used for training non-word-class parameters. After the new word-class labels are re-assigned, the parameters of the ranking module are re-estimated so that they can be used to evaluate the LLR's of various n-grams in the next iteration.

Since the initial class labels assigned to the n-grams to train the ranking module is solely based on the system dictionary, the initial LLR's may not be estimated reliably enough to reflect the true ranks of the n-grams, especially for n-grams which have neither very high nor very low LLR's. Therefore, it would be safer to use only a small percentage of them to update the system, so that the real new words are less likely to be truncated and the recall can be kept high; spurious words are less likely to be assigned the `word-class' label, thus keeping precision high. Currently, we truncate very unlikely candidates among the worst 5% of the candidates as determined by the ranking module and increase the size of the word-class members by 5% per iteration. The effects of adjusting such percentages will be surveyed in our future work. Since the fractions are small, the corresponding sublists will consist almost entirely of words or entirely of non-words. As a result, the system will be refined iteratively without introducing considerable errors.

5.3 Performance Evaluation on the Integrated System

Table 4 shows the estimated performance of the word segmentation module of the iteratively integrated system based on the new word lists extracted from the 1,000 sample sentences.

n-gram	iteration number	p (%)	r (%)	WPR	FM
2	1*	68.67	76.67	72.67	72.45
	21	72.39	80.83	76.61	76.38
	Difference	3.72	4.16	3.94	3.93
3	1*	28.81	80.68	54.75	42.46
	21	38.60	87.80	63.20	53.62
	Difference	9.79	7.12	8.45	11.16
4	1*	38.24	93.45	65.85	54.28
	21	56.21	93.82	75.01	70.30
	Difference	17.97	0.37	9.17	16.03

***Table 4.** Performance of the word segmentation module in the iteratively integrated system.('Difference': difference in performance between iteration 21 and iteration 1.) (*Note: the performance of iteration 1 corresponds to the output of the word segmentation module in the non-iterative scheme.)*

The performance corresponding to iteration 1 is the performance before the augmented dictionary is iteratively refined; in other words, it is the performance of the segmentation module of the non-iterative scheme. Table 4 shows that the recalls for the embedded *new words* are as high as 81%, 88% and 94%, respectively, for 2-, 3-, 4-character words after 21 iterations. This means that more than 81% of the new words will be extracted from the text corpus. Only a small portion of them (6-19%) will be lost. This table also shows that the output new word lists have precision rates of 72%, 39%, and 56%, which are reasonably high for lexicography applications. The joint performance for new word identification, in terms of WPR, is 77%, 63% and 75%. The F-measures are 76%, 54% and 70%, respectively. With this performance, it is believed that the proposed integration method can be a good tool for lexicographers to use to identify new words with considerable savings in human effort.

The effect for iteratively updating the augmented dictionary is shown in the rows entitled 'Difference' in Table 4, which are the differences in performance between the 21st iteration and the first iteration (i.e., non-iterative performance). Note that *both* precision and recall (for all n-grams) are improved at the same time, as expected. The improvement obtained by incrementally truncating the vocabulary of the initial augmented dictionary and by re-maximization of the integrated system is significant, especially for 3-character and 4-character words. As can be seen in the above tables, the precision is improved by about 4% (bigram), 10% (trigram), and 18% (quadragram). The recall rate is improved by about 4% (bigram), 7% (trigram), and 0.4%(quadragram). (The improvement in recall for quadragram new words is small since more than 93% of them have been identified and the number of quadragram new words is smaller than that of bigram or trigram new words.) The improvement in WPR is 4%, 8%, and 9% for bigrams, trigrams and quadragrams, respectively; and the F-measure is improved by 4% (bigram), 11% (trigram), and 16% (quadragram), respectively.

To show whether the smaller improvement for bigrams is statistically significant [Wonnacott 90], a hypothesis test was conducted to determine whether the difference between before and after applying the iterative refinement processes is really significant (see the appendix for the hypothesis test conducted). According to the testing statistics, we can conclude, with more than 99.8% confidence, that the smaller improvement for bigrams is due to adoption of the iterative scheme, not due to sampling variation in estimating the performance (see also the error analysis sections for further quantitative analyses).

Figure 5 shows the change in precision and recall for bigram words in each iteration. The performance increases almost monotonically while low LLR candidates are progressively removed from the vocabulary. Therefore, the proposed method provides a way to stably improve precision and recall without the performance of one offsetting that of the other.

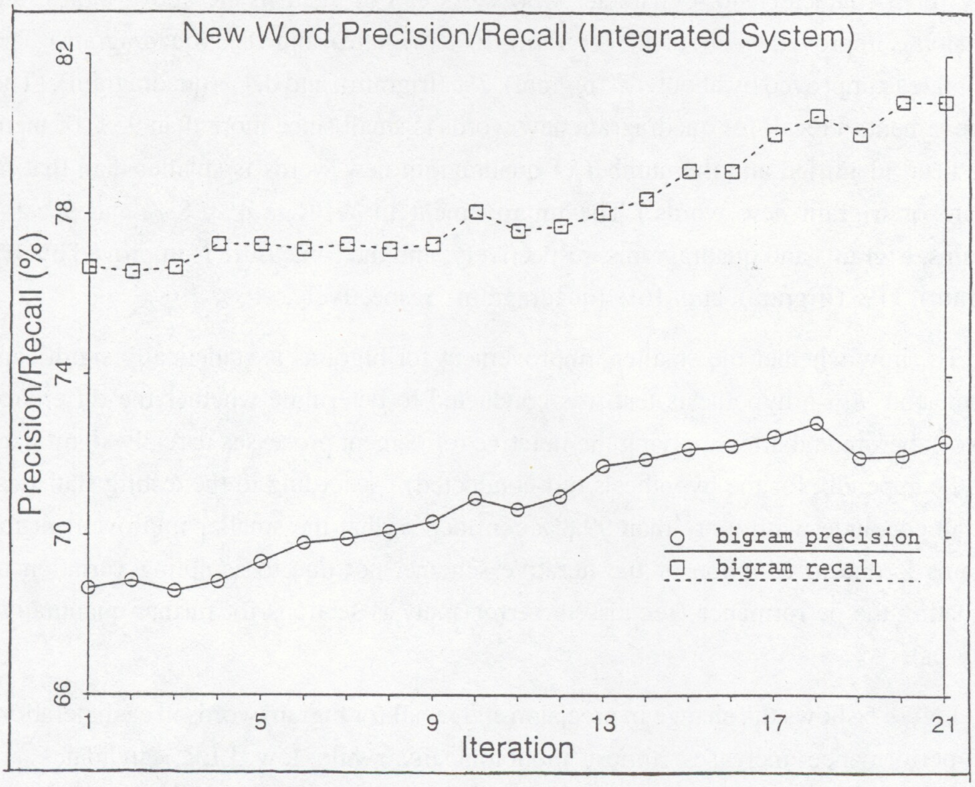


Figure 5 Performance in identifying bigram new words in each iteration.

As for the computation load, each iteration of the integrated system takes about 6 hours of processing time using 2 Sun Sparc10 workstations and one IBM RS/6000 workstation concurrently (through a heavily loaded network with a large number of I/O operations). This processing time is about one the same order as that obtained in [Tung 94] (by scaling up the corpus size to the same amount). According to the *precision* rates in Table 4 and the *sizes* of the output word lists of the segmentation module in the final iteration, it is estimated that we can extract about 12,600 bigram new words, 8,400 trigram new words and 8,500 quadragram new words, i.e., 29,500 new words in total from the 311,591 sentences, including compound words and proper names. Considering the cost of manually identifying such a large corpus, this computer processing time is quite affordable. Furthermore, since the estimated new word recalls do not change significantly near the final iteration and only the precision of the trigrams has a more visible change near the final iteration, we therefore terminate the process at the 21st iteration without waiting for true convergence. Hence, some of the above conclusions may not be extrapolated to later iterations. The percentages for truncating unlikely n-grams from the augmented dictionary can be adjusted to affect the convergence speed.

However, the effect of varying this factor is beyond the scope of this work and will not be addressed here.

5.4 Quantitative Analyses for Performance Improvement

There are several factors which affect the precision and recall rates of the system. The following analysis will further justify our strategy for improving the system performance by iteratively truncating unlikely candidates from the augmented dictionary. Let n_{ww} be the number of true new words that are in the output new word list, let n_{wx} be the number of true new words that do NOT appear in the output new word list, and let n_{xw} be the number of non-word n-grams that are in the output new word list. (The first subscript in the symbols indicates whether an n-gram is a real new word or not, and the second subscript indicates whether the n-gram is identified by the system as a new word or not, where 'w' means 'word' and 'x' means 'non-word'.) The new word precision and recall rates as defined previously are then equivalent to:

$$p = \frac{n_{ww}}{n_{ww} + n_{xw}} = \frac{1}{1 + n_{xw}/n_{ww}}$$

$$r = \frac{n_{ww}}{n_{ww} + n_{wx}} = \frac{1}{1 + n_{wx}/n_{ww}}.$$

Table 4 shows that most of the contributions of the F-measure and WPR comes from the improvement in precision, especially for trigrams and quadragrams. This improvement could come from an increasing n_{ww} or a decreasing n_{xw}. Further detailed investigation shows that n_{ww} (i.e., the number of correctly identified new words) *increases* by 5% for bigrams and by 8% for trigrams between the initial and final iterations; and n_{ww} is almost constant throughout all the iterations for quadragrams in the sample sentences (since 93% of the quadragrams were already identified in the first iteration). On the other hand, n_{xw} (namely, the number of non-words that are incorrectly identified as words) *decreases* by 12%, 30%, and 52% for bigrams, trigrams and quadragrams, respectively.

Since the decrease in n_{xw} is much larger than the increase in n_{ww}, the improvement in precision is mostly attributable to the decrease in n_{xw}. In other words, truncating unlikely candidates from the augmented dictionary is the major factor in improving the performance of the system. This behavior justifies well our previous arguments that the performance can be improved by truncating unlikely candidates from the augmented dictionary.

Furthermore, the truncated n-grams will be replaced by other likely shorter segments (such as by changing '將前往' into '將 前往') or cause other combinations of likely segments to appear in the best segmentation (such as changing '凱悅大 飯店' into '凱悅 大飯店', where the deletion of '凱悅大' cause the better segmentation '凱悅 大飯店' to emerge). In more complicated cases, successive removal of unlikely n-grams will be conducted, such as changing '移送 台中少年 法庭審理' into '移送 台中 少年 法庭審理' and then into '移送 台中 少年法庭 審理', as explained in the Introduction. Therefore, n_{ww} will also increase (and thus improve the precision further) due to the use of association metrics (which reject unlikely candidates that are preferred by the word segmentation module.) Furthermore, because $n_{ww}+n_{wx}$ is a constant (corresponding to the number of real new words in the sample sentences), the increase in n_{ww} also corresponds to the decrease (by the same amount) in n_{wx}, namely, the number of new words that are incorrectly identified as non-words. As a result, the recall rate is also improved due to the increase of n_{ww} and the decrease of n_{wx}. Thus, the precision and recall rates increase almost monotonically.

6. Quantitative Analyses of the Acquired Words

6.1 Distribution of Unknown Words

The following table shows some randomly sampled new words in the output list of the 1,000 sample sentences. They were classified roughly according to [Wang 95] (which will be detailed in later quantitative analyses.)

Bigram New Words		Trigram New Words		Quadgram New Words	
Proper Names					
鹿谷	Lu-Gu; a county name	中新社	China News Service	曾蔡美佐	a female name
蓋茲	(Bill) Gates	富士通	Fujitsu	新興分局	Hsin-Hsing police office
住友	a company name	翁秀卿	a female name	富岡國小	Fu-Gang Primary School
Ordinary Words					
護法	guard	管理局	Bureau of Administration	年度預算	annual budget
幹員	talented (police) men	養豬戶	pig-raising farmers	全球股市	global stock markets
鑑於	in view of	下半年	second half of the year	貨幣市場	monetary market
共舞	dance (with somebody)	投機風	opportunism	國家公園	national park
責令	command	收盤價	closing price	生命安全	personal security
Abbreviation					
市警	city policemen	國台辦	Taiwan-Affair Office of National Affair House	省都委會	provincial city development committee
中菲	Sino-Philippine	消基會	the Consumer Protection Committee	紅會人員	the Red Cross staffs
鄉代	county representatives	上下班	go-to-and/or-come-back-from the office	投開票所	polls
Collocational Strings					
就會	will then ...	據指出	it was indicated that ...	絕大多數	overwhelming majority
既非	neither	並沒有	do not	一片混亂	a mess
Derivational Words					
廠方	authority of the company	壽險業	life insurance companies	所有權人	owner
		複雜化	complicate		
Numerical Strings					
一萬	ten thousands	十四日	14th day of the month	八十年度	1991 accounting year

Table 5. Examples of new words acquired in the 1,000 sample sentences at the final iteration.

To provide a quantitative indication on what kinds of new words are extracted and what kinds of new words are incorrectly identified, the new word list of the 1,000 sample sentences at the final iteration were sampled and classified according to [Wang 95] into the following categories: proper names (P), abbreviational words (A), derived (derivational) words (D), collocational strings (C), numerical strings (#) and other newly coined words (O) (i.e., ordinary new words not generated through the special word formation mechanisms of the P, A, D, C, # categories). 100 samples were drawn from each n-gram new word list derived from the 1,000 sample sentences to estimate their distribution (except two lists which have fewer than 100 tokens in the 1,000 sentences).

The P, A, D, C type words represent new words that were generated from a few productive new word formation processes used by native Chinese people [Wang 95]. Other new words, which were not produced by such special mechanisms were categorized as the O (ordinary or others) type. In [Wang 95], P, A, D, C as well as 'newly coined words', and 'ambiguous words' were used for classification; however, the last two categories could not be easily distinguished, so we classified them and other unclassifiable new words into the O category.

The collocational strings are character strings that are frequently used at the same time in Chinese text (such as '即告' (in the context of '即告失敗'), and '就會'). Some of them may be further divided into shorter words. However, they are so frequently used that Chinese lexicographers may want to include such lexical units into the dictionary ([Wang 95]). One reason is that their meanings may not be easily acquired through certain lexical rules or morphological rules. Such entries may also be arguable for different lexicographers since the qualification heavily depends on the lexicographer's criteria. For instance, the collocational strings listed in Table 5 (and examples such as '乃於' and '一定是' given in [Wang 95]) may not be unanimously approved by all lexicographers. However, the percentages of such debatable strings that are recognized as words are small, as shown in Table 6, especially for trigrams and quadragrams in the current work. Furthermore, various approaches were tested against the same benchmark prepared by the same person, as described in Section 2.3. Therefore, the differences in performance among various approaches very likely reflect the true situation.

The numerical strings were also classified into one special category for quantitative analyses because we didn't use any special treatment on the numerical strings in the current work and, thus, could make mistakes with such strings. As we shall see in later quantitative analyses, numerical strings also contribute a large portion of the errors since they rarely follow the contextual constraints or the association metrics used in the current work. However, the syntax for numerical strings is regular; there is no doubt that such strings can be parsed out using a finite state machine to improve the reported performance further. Therefore we will not comment on this category.

n-gram	P(%)	A(%)	D(%)	C(%)	O(%)	#(%)	Total (%)
2	9	2	0	16	67	6	100
3	34	5	21	7	23	10	100
4	5	4	1	5	82	3	100

Table 6. *Distribution of correctly identified words (P: proper names, A: abbreviational words, D: derived words, C: collocational strings, O: other ordinary new words [Wang 95]).*

Table 6 shows the distribution of various kinds of new words that were correctly identified. The gray cells in the table identify special categories that have larger percentages. The bigram collocational strings (C), 3-character proper names (P) and 3-character derived words (D) are particularly noticeable except for the large numbers of ordinary new words (i.e., O-type new words). The high percentage of 3-character proper names is due to the fact that most Chinese proper names are 3 characters long. Therefore, the current approach provides a good tool for extracting proper names. As for quadragrams, the largest portion is attributable to compound words which consist of two double-character words. The bigram O-type new words, on the other hand, are hard to classify according to special word formation rules since bigram Chinese words are most numerous among Chinese words, and it appears that they belong to many different syntactic variants.

6.2 Error Analyses

There are two types of errors in extracting new words. The first type occurs when real new words are missed (referred to as Type I errors), and the second type occurs when non-words are recognized as new words (referred to as Type II errors). In the 1,000 sample sentences, 1099 errors occurred; Type I errors resulted in 219 missed n-grams (76% of which were bigrams, 16% were trigrams and 8% were trigrams); Type II errors resulted in 880 mis-recognized non-word n-grams (including 30% bigrams, 47% trigrams, and 23% quadragrams). Therefore, about 80% of the errors were attributable to the Type II errors. This means that we can still truncate more spurious words from the augmented dictionary to achieve better performance if we conduct more iterations. Furthermore, since most Type I errors are attributable to bigrams, we can clearly see, from Table 4, that trigram and quadragram words were recognized with a very high recall rate (88% and 94%). A few errors of the two types are listed as follows to illustrate:

Type I Errors:

拖吊 (towing), 除權 (split of stock shares), 營收 (income), 分案 (filing another case), 天價 (a historical price), 工寮 (worker's ouses), 代金 (substitute money), 平均線 (moving average), 任立渝 (a weather reporter's name), 副都心 (alternative city center), 雙黃線 (double yellow lines (for delineating car lanes)), 中央銀行 (the Central Bank), 工業局長 (Chief Officer of the Bureau of Industry), 無償配股 (unpaid stock shares), 新聞媒體 (news media)

Type II Errors:

將是 (will be), 已將 (have been), 爲此 (for this purpose/reason), (高) 黃秀 (a female name), (黃) 文忠 (a male name), 應朝 (should ... toward), 並向 (and ... from), (漲少) 跌多 ((rarely appreciated but) frequently depreciated), 之二是 (two of the ... are), 與台灣 (... with Taiwan), 目前正 (currently is doing something ...), 其二哥 (his secondary elder brother), 的現象 (the phenomena of ...), 的拖吊 (... towing), 各自的 (individual), 除權後 (after splitting), 額滿後 (after fully registered), 作業中 (during operation), 狀況下 (under ... situation), 將下降 (will decrease), 國際標準 (組織) (ISO), 上週大盤 (the closing stock indices of the last week), 無法接受 (cannot accept), (鄉) 代會主席 (chair of the committee of the county citizen representatives), 住基隆市 (live in Keelung City), 教育局說 (the officers of the Bureau of Education said).

(Note: the Chinese characters within the parentheses are those strings which must be patched with the characters not in parentheses in order to form a non-spurious phrase.)

The distribution of such errors and analyses of the errors are outlined as follows.

Table 7 shows the estimated percentages of missing new words. (Note that the number of such trigrams (36 entries) and quadragrams (17 entries) is less than 100 samples in both cases; therefore, the estimated percentages for trigrams and quadragrams are not as reliable as the other estimates.) Among the unrecognized trigram and quadragram new words, most are proper names (or numerical strings); therefore, by improving proper name recognition a little further, the proposed method can be expected to achieve even higher recall rates for trigram and quadragram words. (Table 8, discussed below, further justifies this statement since Type II errors for trigrams and quadragrams that originated from proper names are not the major source of Type II errors.) The Type I errors for bigrams are evenly distributed among the P (proper name), C (collocational string), and O (other ordinary new word) types. Examining a few instances of such un-recognized bigrams shows that some of the bigrams are embedded in spurious trigrams having extra special characters (e.g., '(的) 拖吊 ' and ' 除權 (後)') or quadragram collocational strings which consist of two double-character words (like

'(七月)營收 '). This may suggest that such spurious trigrams and quadragrams (as will be examined in Table 8) should be further truncated to bring those un-recognized bigrams back to the best segmentation.

n-gram	P(%)	A(%)	D(%)	C(%)	O(%)	#(%)	Total (%)
2	13	6	4	26	37	14	100
3	25	8	5	3	17	42	100
4	24	5	0	0	24	47	100

Table 7. *Distribution of missing new words (Type I Error) (P: proper names, A: abbreviational words, D: derived words, C: collocational strings, O: other ordinary new words, #: numerical strings).*

Table 8 further shows the distribution for Type II errors, in which non-word n-grams (i.e., spurious words) are classified as words. The column titles, P, A, D, C, O and #, now refer to the categories of the major substrings of such non-word n-grams. For instance, the P column shows the percentage of mis-recognized non-word strings which originated mainly from proper names.

n-gram	P(%)	A(%)	D(%)	C(%)	O(%)	#(%)	Total (%)
2	25	0	0	47	12	16	100
3	5	0	0	13	59	23	100
4	10	3	2	41	20	24	100

Table 8. *Distribution of spurious words that were recognized as words (Type II Error) (P: originated from Proper names, A: originated from abbreviational words, D: originated from derived words, C: originated from collocational strings, O: originated from other ordinary words, #: originated from numerical strings.)*

The largest portion of the bigram spurious words came from collocational strings whose individual characters often appeared at the same time. Most (44 out of 47) such collocational strings need to be further divided into two single-character words to fully decompose them into the smallest meaningful units (e.g., ' 將是 ', ' 已將 ', ' 爲此 ').

However, since they are frequently used at the same time, even native Chinese speakers may be confused about whether such strings should be considered as single lexical units. (As described earlier, some of these collocational strings were indeed considered as words or lexical units, and were included in the statistics in Table 4.) Therefore, it depends more or less on the lexicographer's opinion how useful it is to include such strings in the lexicon. The second largest source of bigram spurious words was the substrings of the 3-character proper names (e.g., (高) 黃秀 (a female name), (黃) 文忠 (a male name)). The remaining bigram spurious words were ordinary words derived by over splitting longer words. For instance, ' 跌多 ' ('frequently depreciated') was acquired by splitting the preferred idiomatic expression ' 漲少跌多 ' ('rarely appreciated and frequently depreciated').

For trigrams, most spurious words originated from ordinary words. Further investigation shows that most of the trigrams are composed of an ordinary word and a special character. For instance, 27 n-grams out of the 59 O-type instances contain the ' 的 (de)' character as the first or the last character of the trigram (e.g., ' 的拖吊 ', ' 各自的 '). Other special characters include time and location markers, such as ' 後 ' ('after...', e.g., ' 除權後 ', ' 額滿後 '), ' 中 ' ('during ...', 'within...', e.g., ' 作業中 '), and ' 下 ' ('under... situation', 'beneath...', e.g., ' 狀況下 '), at the end of the trigram, or ' 將 ' ('will') at the beginning of the trigram (e.g., ' 將下降 '). To resolve such errors, it would be a good idea to include a mechanism for removing n-grams that contain such special characters. The collocational strings also contribute a large portion of the spurious words, which are formed by concatenating single-character words and double-character words that frequently co-occur due to syntactic collocation (e.g., ' 與台灣 ', ' 目前正 ').

The spurious quadragrams, shown in the last row, consist mainly of collocational strings and ordinary n-grams. Both kinds of spurious words share the same feature in that they consist mostly of two bigram words. The portion of spurious words corresponding to the 'O' column contains many 'noun+noun' (or 'adjective+noun') pairs. Although they take the form of compound words, they may not be considered sufficiently useful for the lexicographers to include such quadragrams into the new word dictionary (e.g., ' 上週大盤 ' ('the closing stock indices of the last week')); thus, they are considered `incorrectly identified' when the system suggests so. Furthermore, some spurious quadragrams are substrings of longer compound words (e.g., ' 國際標準 ' is considered a substring of ' 國際標準組織 ' (ISO)), so they are not regarded as words. The collocational strings have more syntactic variants, which assume the form of 'auxiliary verb+verb', 'verb+noun', 'noun+verb', and so on. Therefore, it may not be easy to reject such spurious words without using some syntactic information in the model.

To sum up, it is probably helpful to apply a numerical string pre-processor before the current approach is applied. Among the remaining errors, Type II errors were the most common type for the current approach (which resulted in worse precision performance in comparison to the recall performance). There was still a small fraction of proper names which could not be resolved completely. Furthermore, to improve the precision of the n-grams, it might be useful to detect the existence of some special characters (such as '的' and '將') within the n-grams in order to remove such spurious words. It would also probably be useful to include additional syntactic processing steps in the filtering stage, so that spurious quadragram words which consist of two double-character words could be removed.

7. Concluding Remarks and Future Research

An unsupervised method is proposed for extracting Chinese new words from a large text corpus in this paper. It provides an architecture for improving both the precision and recall systematically. The proposed method avoids certain drawbacks of the general segmentation-merging-filtering- disambiguation scheme by iteratively integrating the word segmentation module and the likelihood ratio ranking module, so that they can enhance one another's discrimination power. The segmentation task is conducted using an *augmented dictionary*, which contains *potential unknown word candidates* in the input corpus in addition to known words in the system dictionary. The unknown words are extracted from the segmented corpus after the corpus is processed by a statistical word segmenter. The set of potential unknown words is then progressively refined using the joint character association metric provided by the likelihood ratio ranking module, so that the next segmentation iteration can be conducted based on a better set of potential unknown words. Accordingly, both the contextual information (which maximizes the likelihood of the text corpus in terms of known and unknown words) and the intrinsic association features of the n-grams are iteratively integrated to supplement each other. Therefore, n-grams which satisfy the contextual constraints imposed by the word segmentation module can be further justified using the joint association metrics provided by the likelihood ratio ranking module, and the ambiguity between overlapping candidates generated by the segmentation-and-merging scheme is implicitly resolved by applying the contextual constraints in the next segmentation session. The parameters of the ranking module can also be improved by using the progressively refined output word list of the segmentation module to re-assign progressively more appropriate class labels to the corpus n-grams, thus providing better estimates of the log-likelihood ratios for truncating inappropriate word candidates. In this iterative process, the precision is progressively improved through the filtering operations, and the real new words corresponding to the rejected spurious candidates are recalled through re-segmentation. The results show increasingly improved performance both in precision and recall through iterations.

It has been observed that the system can acquire new words with sufficiently high precision in comparison to other previous works and still retain high recall. Since this approach is unsupervised, a substantial amount of human processing effort can be saved, and no pre-segmented training corpus is required before compiling the new word list. The cost of the proposed approach is, thus, very low. Based on the results of this study, it is expected that the proposed framework can provide a good tool for lexicographers to use to extract Chinese new words from a large text corpus.

To summarize, the performance obtained using different strategies is listed in the following table. It is easily seen that the iterative approach, which integrates the information used in both the word segmentation module and the likelihood ranking module, outperforms the non-iterative scheme and the other two models, which use only contextual information or association features.

n-gram	Models	p (%)	r (%)	WPR	FM
2	WS-only	68.72	78.41	73.57	73.25
	LRRM-only	54.28	90.99	72.63	68.00
	Non-Iterative	73.56	73.90	73.73	73.73
	Iterative	72.39	80.83	76.61	76.38
3	WS-only	29.63	81.36	55.50	43.44
	LRRM-only	33.78	63.01	48.40	43.98
	Non-Iterative	31.90	80.34	56.12	45.66
	Iterative	38.60	87.80	63.20	53.62
4	WS-only	38.96	93.09	66.03	54.93
	LRRM-only	51.17	81.42	66.30	62.84
	Non-Iterative	42.38	93.09	67.74	58.25
	Iterative	56.21	93.82	75.01	70.30

Table 9. *Comparison of performance between various models for new word extraction. (WS-only: Word-segmentation only, LRRM-only: ranking module only, Non-Iterative: cascading the WS and LRRM modules and truncating the worst 10% words in the segmentation output; Iterative: Iteratively integrating WS and LRRM modules.)*

Finally, in the current work, the contextual constraints and other association metrics are integrated by iteratively applying them to extracting likely words and removing unlikely word candidates. However, it will be more elegant if such information can be closely integrated by including both the contextual constraints and other global character association measures in the same model, so that the refined language model for word segmentation directly simulates lexicographers' sense of word formation more closely. We can even include some known lexical and syntactic constraints on the Chinese lexicon in a probabilistic framework, such as by including 'the probability that a word will contain a 'de' marker' (which should be close to zero) in the model. Such integration will then bring the segmentation model even more in line with Chinese word formation rules, and thus allow us to remove strings such as '的人' in a more elegant way, without resorting to special processing. This issue will be left for future research.

Appendix: Statistical Significance Test on Performance Improvement

To see whether the smaller improvement for the bigrams in the iteratively integrated system is statistically significant [Wonnacott 90], we divided the 1,000 sample sentences into 10 groups (with 100 sentences per group) and estimated their respective performance (i.e., precision, recall, WPR and FM). The differences in performance, $D_i \equiv X_i - Y_i$ (i=1, 10), between the first and the final iterations are estimated (X_i: performance for the *i*-th group of sentences after iteratively refining the augmented dictionary; Y_i : performance before applying iterative refinement). The average difference ($\bar{D} \equiv \bar{X} - \bar{Y}$) was then compared with the sample variance (s_D^2) of the differences to see whether such differences are statistically significant. More precisely, we carried out a hypothesis test to reject the hypothesis that there is no difference in performance (i.e., $\mu_D = 0$) between the first and the final iterations based on the following testing statistic:

$$T = \frac{D - \mu_D}{\sqrt{\sigma_D^2/K}}$$

$$D = \frac{1}{K} \sum_{i=1}^{K} D_i \qquad (K = 10)$$

$$D_i = X_i - Y_i \qquad (i = 1, K)$$

$$\sigma_D^2 \approx s_D^2 = \frac{1}{K-1} \sum_{i=1}^{K} \left(D_i - D\right)^2,$$

where D is the random variable denoting the difference in performance, $\bar{D}$ is the sample mean of all the estimated differences D_i (for the *i*-th group of sentences), μ_D is the population mean of the difference,

σ_D^2 is the population variance, and K is the number of groups used in estimating the differences (K=10 in the current test). In the above test, σ_D^2 is approximated by its sample variance s_D^2, and D_i is assumed to be independently and identically distributed with mean μ_D and variance σ_D^2. With K=10, T can be approximated as a student-t distribution with 9 degrees of freedom. To reject the hypothesis with more than 95% confidence, the T value must be greater than 1.83 when the hypothesis holds (i.e., when $\mu_D = 0$). In fact, we have T= 4.248, 9.142, 7.439, and 7.436 for bigram precision, recall, WPR and FM, respectively, which all exceed the threshold 3.89 for 99.8% confidence. We can, therefore, conclude that the smaller improvement for bigrams is not due to the statistical variation in estimating the performance, but due to the effect of the proposed iterative algorithm.

References

Appelt, Douglas E., Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson, "FASTUS: A Finite-State Processor for Information Extraction from Real-World Text," *Proc. IJCAI-93*, Chambery, France, Aug. 1993.

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

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, Jing-Shin, Yi-Chung Lin and Keh-Yih Su, "Automatic Construction of a Chinese Electronic Dictionary," *Proceedings of the Third Workshop on Very Large Corpora*, pp. 107-120, MIT, June, 1995.

Chang, Jing-Shin and Keh-Yih Su, "A Multivariate Gaussian Mixture Model for Automatic Compound Word Extraction," to appear in *Proceedings of ROCLING X*, ROC Computational Linguistics Conferences, Academia Sinica, Taipei, Taiwan, 1997. (also in Chang, Jing-Shin, *Automatic Lexicon Acquisition and Precision-Recall Maximization for Untagged Text Corpora*, PhD dissertation, Department of Electrical Engineering, National Tsing-Hua University, Taiwan, 1997.)

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.

Chen, S.-C., J.-S. Chang, J.-N. Wang and K.-Y. Su, "ArchTran: A Corpus-Based Statistics-Oriented English-Chinese Machine Translation System," *Proceedings of Machine Translation Summit III*, pp. 33-40, Washington, D.C., USA, July 1-4, 1991.

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.

Chiang, T.-H., Y.-C. Lin and K.-Y. Su, "On Jointly Learning the Parameters in a Character-Synchronous Integrated Speech and Language Model," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 3, pp. 167-189, May 1996.

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

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

Devijver, Pierre A. and Josef Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall Inc., N.J., USA, 1982.

Duda, Richard O. and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, NY, USA, 1973.

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.

Fung, Pascale and Dekai Wu, "Statistical Augmentation of a Chinese Machine-Readable Dictionary,"*Proceedings of the Second Annual Workshop on Very Large Corpora*, pp. 69-85, Kyoto, Japan, August, 1994.

Hirschman, Lynette and Marc Vilain, *Extracting Information from the MUC*, Tutorial of the ACL 95, MIT, Cambridge, MA, June 16, 1995.

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

Hobbs, Jerry R. "FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text," *Proc. of ROCLING IX*, pp. 199-231, Natl. Cheng-Kung Univ., Tainan, Taiwan, Aug. 1996.

Huang, C.-R., K.-J. Chen, L.-P. Chang and H.-L. Hsu, "Introduction to the Academia Sinica Balance Corpus (中央研究院平衡語料庫簡介)," *Proceedings of ROCLING VIII*, pp. 81-99, Taiwan, ROC, 1995.

Juang, Biing-Hwang, and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Signal Processing*, vol. 40, no. 12, pp. 3043-3054, Dec. 1992.

Lin, M.-Y. A Study in Chinese Word Segmentation, *Master Thesis*, National Tsing-Hua University, Taiwan, ROC, 1993.

Lin, M.-Y., T.-H. Chiang, and K.-Y. Su, "A Preliminary Study on Unknown Word Problem in Chinese Word Segmentation," *Proceedings of ROCLING VI*, pp. 119-141, Taiwan, ROC, 1993.

Papoulis, A., *Probability & Statistics*, Prentice Hall, Inc., Englewood Cliffs, NJ, USA, 1990.

Rabiner, L., and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1993.

Roussas, G. G., *A First Course in Mathematical Statistics*, Addison-Wesley Publishing Company, 1973.

Salton, Gerard and Michael J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

Smadja, Frank, "Retrieving Collocations from Text: Xtract," *Computational Linguistics*, vol. 19, no. 1, pp. 143-177, 1993.

Smadja, Frank, Kathleen R. McKeown and Vasileios Hatzivassiloglou, "Translating Collocations for Bilingual Lexicons: A Statistical Approach," *Computational Linguistics*, vol. 22, no. 1, pp. 1-38, 1996.

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.

Sproat, R., C. Shih, W. Gale and N. Chang, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese," *Computational Linguistics*, vol. 22, no. 3, pp. 377-404, 1996.

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, 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.

Wang, Mei-Chu, Chu-Ren Huang and Keh-Jiann Chen, "The Identification and Classification of Unknown Words in Chinese: An N-Grams-Based Approach," In Ishikawa, Akira and Yoshihiko Nitta, Eds. Festschrift for Professor Akira Ikeya, pp. 113-123. Tokyo: The Logico-linguistics Society of Japan, 1995.

Wonnacott, Thomas H. and Ronald J. Wonnacott. *Introductory Statistics (5th Ed.)*, John Wiley & Sons, Inc. NY, USA. 1990.

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, 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.