STATISTICAL MODELS FOR WORD SEGMENTATION AND UNKNOWN WORD RESOLUTION

Tung-Hui Chiang, Jing-Shin Chang, Ming-Yu Lin and Keh-Yih Su

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
{andy,shin,felipe,kysu}@ee.nthu.edu.tw

ABSTRACT

In a Chinese sentence, there are no word delimiters, like blanks, between the "words". Therefore, it is important to identify the word boundaries before processing Chinese text. Traditional approaches tend to use dictionary lookup, morphological rules and heuristics to identify the word boundaries. Such approaches may not be applied to a large system due to the complicated linguistic phenomena involved in Chinese morphology and syntax. In this paper, the various available features in a sentence are used to construct a generalized word segmentation model; the various probabilistic models for word segmentation are then derived based on the generalized model.

In general, the likelihood measure adopted in a probabilistic model does not provide a scoring mechanism that directly indicates the real ranks of the various candidate segmentation patterns. To enhance the baseline models, a robust adaptive learning algorithm is proposed to adjust the parameters of the baseline models so as to increase the discrimination power and robustness of the models.

The simulation shows that cost-effective word segmentation could be achieved under various contexts with the proposed models. It is possible to achieve accuracy in word recognition rate of 99.39% and sentence recognition rate of 97.65% in the testing corpus by incorporating word length information to a context-independent word model and applying a robust adaptive learning algorithm in the segmentation process.

Since not all lexical items could be found in the system dictionary in real applications, the performance of most word segmentation methods in the literature may degraded significantly when unknown words are encountered. Such an "*unknown word problem*" is also examined in this paper. An error recovery mechanism based on the segmentation model is proposed.

Preliminary experiments show that the error rates introduced by unknown words could be reduced significantly.

1. Introduction

Most natural language processing tasks, such as machine translation or spoken language processing, take *words* as the smallest meaningful units. However, no obvious delimiter markers can be observed between Chinese words except for some punctuation marks. Therefore, word segmentation is essential in almost all Chinese language processing tasks. (The same is true for other languages like Japanese.)

Matching input characters against the lexical entries in a large dictionary is helpful in identifying the embedded words. Unfortunately, an input sentence can usually be segmented into more than one segmentation patterns. For example, a Chinese sentence like:

對方姑娘而言，立志當政治家的沒有一個功成名就的。

may include the following ambiguous segmentation patterns based on simple dictionary lookup:

1.+ 對 方姑娘 而言 ， 立志 當 政治家 的 沒有 一 個 功成名就 的。
TO MS. FANG, those who decide to BE A STATESMAN never succeed and become famous.

2.* 對 方姑娘 而言 ， 立志 當政 治家 的 沒有 一 個 功成名就 的。
TO MS. FANG, those who decide to HOLD POWER and MANAGE A HOUSEHOLD never ...

3.* 對方 姑娘 而言 ， 立志 當政 治家 的 沒有 一 個 功成名就 的。
To the LADY of the COUNTER PARTY, those who decide to HOLD POWER and MANAGE A HOUSEHOLD never ...

4.* 對方 姑娘 而言 ， 立志 當 政治家 的 沒有 一 個 功成名就 的。
To the LADY of the COUNTER PARTY, those who decide to BE A STATESMAN never ...

⋮

where the first segmentation pattern is the preferred one. To find the correct segmentation pattern, it is necessary to use other information sources in addition to dictionary lookup. The main issue for dealing with the word segmentation problem is how to find out the *correct* segmentation from all possible ones.

There are several technical reasons that make the word segmentation problem nontrivial. First, the Chinese characters can be combined rather freely to form legal words. As such, ambiguous segmentation patterns may not be resolved by using simple dictionary lookup.

Second, a Chinese text contains not only words but also inflectional or derivational *morphemes*, *tense markers*, *aspect markers*, and so on. Because such morphemes and markers may often be combined with adjacent characters to form legal words as well as standing alone as a word, it is hard to deal with such ambiguities with simple morphological analysis.

Third, *unknown words* may appear in the input text. This fact may make many word segmentation models work badly in real applications, because most segmentation algorithms today assume that all words in the input text could be found in the system dictionary. In fact, unknown word resolution has become the major bottleneck with the current segmentation techniques.

To resolve these problems, various knowledge sources might have to be consulted. However, extensive use of high level knowledge and analysis may requires extremely high computation cost. Hence, segmentation algorithms that make use of discriminative and easily acquired features are desirable.

In the past, two different methodologies were used for word segmentation; some approaches are *rule-based* (Chen [3, 4], Ho [7], Yeh [10]) while others are *statistical* ones (Chang [2], Fan [6], Sproat [8]). Since it is costly to construct lexical or morphological rules by hand, no objective preference could be given for ambiguous segmentation patterns, and it is difficult to maintain rule consistency as the size of the rule base increases, it is less favorable to use a rule-based approach in large scale applications. On the contrary, as data are jointly considered in a statistical framework, statistical approaches usually do not suffer from the consistency problem. Also, global optimization can usually be modeled in statistical frameworks, rather than local constraints by rules. Therefore, statistical approaches are usually more practical in a large application like machine translation. However, the current statistical approaches usually use a maximum likelihood measure to evaluate preference without regarding to the discrimination power of such models. As a result, when the baseline models introduce errors, heuristic approaches, such as adding special information to the dictionary or resorting to later syntactic or semantic analyses are suggested (Chang [2]) to remedy the modeling and estimation errors. Such approaches not only destroy the uniformity of the statistical methods but also make maintenance difficult.

To resolve the above problems, several probabilistic models are proposed in this paper based on a generalized word segmentation model. The focus is to derive different formulations under different constraints of the available resources. In particular, features that could be acquired inexpensively will be used for cost-effective word segmentation so that deep analyses are needed only to the least extent.

In order to adapt the probabilistic models to reflect the real *ranks* of the candidate segmentation patterns and to suppress *statistical variations* among different application domains, a discrimination and robustness oriented adaptive learning algorithm (Su [9], Chiang[5]) is applied to enhance the performance. Moreover, the *unknown word problem* will be addressed and be examined against the proposed models; some experiment results are given and general guidelines to this problem will be suggested.

2. Word Segmentation Models

2.1 A Generalized Word Segmentation Model

For an input sentence with n Chinese characters $c_1, c_2, \cdots, c_n$ (represented as c_1^n hereafter), it might have several different ways of segmentation according to the system dictionary. The goal of word segmentation is to find the *most probable* segmentation pattern for the given character string. Since a segmentation pattern can be identified uniquely with the sequence of words of the segmented sentence. The goal is equivalent to finding a word sequence

$$\hat{W} \equiv \underset{W_i}{\operatorname{argmax}}\, P\left(W_i \mid c_1^n\right) \tag{2.1.1}$$

with the largest *segmentation score* $P\left(W_i \mid c_1^n\right)$. In this formula, $\underset{W_i}{\operatorname{argmax}}\, P\left(\cdot\right)$ refers to the argument, among all possible W_i's, that maximizes the probabilistic function $P\left(\cdot\right)$, and $W_i \equiv w_{i,1}^{i,m_i} = w_{i,1}, w_{i,2}, \cdots, w_{i,m_i}$ denotes the *i*-th possible word sequence with m_i words, whose *j*-th element is $w_{i,j}$.

In general, we could formulate the segmentation score by involving whatever features that are considered discriminative or available, subject only to the constraints of the complexity of the model and the number of parameters that need to be trained. In particular, we would like to use the segmented words (W_i), the word length information (L_i), the number of characters (n) in the input sentence and the number of words (m_i) for the *i*-th segmentation pattern as the features for word segmentation. ($L_i \equiv l_{i,1}^{i,m_i} = l_{i,1}, l_{i,2}, \cdots, l_{i,m_i}$ refers to the *i*-th sequence of word lengths, where $l_{i,j}$ denotes the length of the *j*-th word in the *i*-th possible

word sequence.) These features could be acquired inexpensively in general. Thus, they are adopted in the current task. With these features, we can identify a "segmentation pattern" uniquely with a (W_i, L_i, m_i) triple, and the goal of word segmentation would become to find the word segmentation pattern corresponding to

$$\underset{i}{\operatorname{argmax}}\, P\left(W_i, L_i, m_i \mid c_1^n, n\right) \tag{2.1.2}$$

Hence, we could define a *generalized segmentation score* as:

$$P\left(W_i, L_i, m_i \mid c_1^n, n\right) \tag{2.1.3}$$

Note that the variables, such as W_i and L_i, are not independent. Technically, however, these features are integrated in a single formula so that all models that are computationally feasible could be derived from this general formula; unavailable features will simply be ignored when deriving a particular model.

The generalized segmentation score can be estimated in several different ways depending on the available information resources. In the following sections, we will give a more detailed derivation of a particular model, which takes advantage of the segmented words and the word length information for segmentation. Other models can be derived in much the same way. So they are simply listed without proof.

2.2 Computational Models for Word Segmentation

Assume that a segmented text corpus is available, then we can use the frequency information of the words and their lengths (in characters) for segmentation. The corresponding segmentation score for the *i*-th segmentation pattern will be:

$$\begin{aligned}
&P\left(L_i, W_i, m_i \mid c_1^n, n\right) \\
&= P\left(l_{i,1}^{i,m_i}, w_{i,1}^{i,m_i}, m_i \mid c_1^n, n\right) \\
&\equiv P_i\left(l_1^m, w_1^m, m \mid c_1^n, n\right) \\
&= P_i\left(l_1^m, w_1^m \mid m, c_1^n, n\right) \times P_i\left(m \mid c_1^n, n\right) \\
&= \prod_{k=1}^{m_i} P_i\left(l_k, w_k \mid l_1^{k-1}, w_1^{k-1}, m, c_1^n, n\right) \times P_i\left(m \mid c_1^n, n\right) \\
&= \prod_{k} P_i\left(l_k \mid w_k, l_1^{k-1}, w_1^{k-1}, \cdots, n\right) \cdot P_i\left(w_k \mid l_1^{k-1}, w_1^{k-1}, \cdots, n\right) \times P_i\left(m \mid c_1^n, n\right)
\end{aligned} \tag{2.2.4}$$

For notational simplicity, $P_i(\cdot)$ is used specifically to denote the probability for the *i*-th segmentation pattern, and all the respective *i* indices are dropped from the equation. The multiplication theory for probability: $P(a, b \mid c) = P(a \mid b, c) \times P(b \mid c)$, is applied repeatedly in the derivation, which results in the product terms, indexed by *k*, in the last two formulae.

Since l_k is unique once w_k is given, we have $P(l_k \mid w_k, \cdots) = 1$ for the first term in the equation. If we assume that the *k*-th word depends only on the length l_{k-1} of the previous word, the second term in the last formula can be approximated as $P\left(w_k \mid l_1^{k-1}, w_1^{k-1}, \cdots, n\right) \approx P(w_k \mid l_{k-1})$. Furthermore, if we assume that the number of words m_i depends only on the length of the sentence n, then we have $P_i\left(m \mid c_1^n, n\right) \approx P_i(m \mid n)$. With these assumptions, the segmentation problem is equivalent to finding:

$$\begin{aligned} &\underset{i}{\operatorname{argmax}} P(W_i, L_i, m_i \mid c_1^n, n) \\ &\approx \underset{i}{\operatorname{argmax}} \prod_k P_i(w_k \mid l_{k-1}) \times P_i(m \mid n) \\ &= \underset{i}{\operatorname{argmax}} \sum_k \log P_i(w_k \mid l_{k-1}) + \log P_i(m \mid n) \end{aligned} \tag{2.2.5}$$

where $\log(\cdot)$ refers to a logarithmic function. (The log-scaled probabilities are used simply to reduce the computation time and avoid mathematical underflow.) There are several variants of the above equation, depending on different assumptions made in deriving the segmentation score. First, it is possible to drop the term $P_i(m \mid n)$ or $\sum \log P_i(w_k \mid l_{k-1})$, depending on what information is available, in the previous derivation steps. Alternatively, we can also assume that the word w_k does not depend on the length of the preceding word length l_{k-1}, and thus use $P_i(w_k)$ instead of $P_i(w_k \mid l_{k-1})$ in the formula. By changing the roles of w_k and l_k in the last step of derivation, we can use the transition probability $P_i(l_k \mid l_{k-1})$ instead of $P_i(w_k \mid l_{k-1})$ in the segmentation score. Therefore, the above formula along with its variants constitute a family of segmentation scores as shown below:

$$\begin{aligned} &\underset{i}{\operatorname{argmax}} P(W_i, L_i, m_i \mid c_1^n, n) \\ &\approx \underset{i}{\operatorname{argmax}} \begin{cases} \sum_{k=1}^{m_i} \log P_i(w_k) & (M1) \\ \sum_{k=1}^{m_i} \log P_i(l_k \mid l_{k-1}) & (M2) \\ \log P_i(m \mid n) & (M3) \\ \sum_{k=1}^{m_i} \log P_i(w_k \mid l_{k-1}) & (M4) \end{cases} \end{aligned} \tag{2.2.6}$$

Model M1 is a context-independent word model. It assumes that all words are independent of the other contextual information. Such a model is used in Chang [2] for the segmentation task.

Model M2 uses only the word length transition probabilities in determining the word segmentation patterns. Model M3, on the other hand, uses the number of characters and the number of words in a sentence as the features for segmentation. It seems that such features have nothing to do with the characteristics of Chinese words. However, as shown in Chang [2] and other literatures, most Chinese words are double-character words, single-character words and tri-character words; more than 99% of Chinese words fall within 4 characters. Hence, it is possible to make guesses based on word length information.

Moreover, the length information could be acquired without much extra cost when preparing a segmented corpus. Therefore, such features could provide an inexpensive way for word segmentation in applications where a large dictionary is not available or expensive to acquire. In fact, as will be seen in the performance evaluation section, the performance of such formulations is comparable with others. So it could be used, for instance, to bootstrap the automatic construction process of an electronic dictionary, where there is not a large dictionary initially.

Model M4 uses both word sequence and word length information for segmentation. If the word length information is ignored, this model reduces to M1. By using the extra word length information, which could be acquired from the same corpus for training model M1, this model could make use of more information and the performance is expected to be better if the training corpus is large enough to provide reliable estimation of the model parameters.

If a sentence is annotated with *lexical tags* (i.e., parts of speech) $T_{i,j} \equiv t_{i,j,1}, \cdots, t_{i,j,m_i}$, then it is possible to use such information to define a modified segmentation score. (Tag $t_{i,j,k}$ stands for the *k*-th part of speech in the *j*-th possible tag sequence of the *i*-th segmentation pattern.) One can achieve the same optimization criteria as that of the generalized segmentation score by noting that:

$$\begin{aligned} & \underset{i}{\operatorname{argmax}}\, P\left(W_i, L_i, m_i \mid c_1^n, n\right) \\ & = \underset{i}{\operatorname{argmax}} \sum_{all\ T_{i,j}} P\left(W_i, L_i, T_{i,j}, m_i \mid c_1^n, n\right) \\ & \approx \underset{i}{\operatorname{argmax}} \left[\max_{all\ T_{i,j}} P\left(W_i, L_i, T_{i,j}, m_i \mid c_1^n, n\right) \right]. \end{aligned} \tag{2.2.7}$$

The last formula means to find the tag sequence $T_{i,j}$ with the largest score as defined by

$$P\left(W_i, L_i, T_{i,j}, m_i \mid c_1^n, n\right) \tag{2.2.8}$$

for each possible segmentation pattern. Then select the segmentation pattern with the highest maximum score as the preferred segmentation pattern.

By following the same procedures as in Eq. (2.2.4) and making some assumptions, it is not difficult to find that the following word segmentation models could be used when the lexical tag information is available:

$$\begin{aligned}
&\underset{i}{\operatorname{argmax}}\, P\left(W_i, L_i, m_i \mid c_1^n, n\right) \\
&\approx \underset{i}{\operatorname{argmax}} \begin{cases}
\max\limits_{T_{i,j}} \sum\limits_{k=1}^{m_i} \log P_{ij}\left(t_k \mid t_{k-1}\right) & (M5) \\
\max\limits_{T_{i,j}} \sum\limits_{k} \log P_{ij}\left(w_k \mid l_{k-1}\right) + \sum\limits_{k} \log P_{ij}\left(t_k \mid t_{k-1}\right) & (M6) \\
\max\limits_{T_{i,j}} \sum\limits_{k} \log P_{ij}\left(w_k \mid t_{k-1}\right) + \sum\limits_{k} \log P_{ij}\left(t_k \mid t_{k-1}\right) & (M7)
\end{cases}
\end{aligned} \tag{2.2.9}$$

Here, we use $P_{ij}(\cdot)$ to specify the probability associated with the *i*-th segmentation pattern and the *j*-th tag sequence, with the corresponding indices within the parentheses omitted.

Model M5 is used to find the best parts of speech sequence associated with the ambiguous segmentation patterns. So the segmentation pattern that produces the most possible lexical tag sequence is regarded as the desired one. In Model M6, the parts of speech sequence is taken into account to facilitate word segmentation model M4. In model M7, the segmentation is considered best if the segmentation pattern maximizes the sequence of corresponding parts of speech and the sequence of words. Because both word sequence and lexical tag sequence are the target of optimization in this process, such a formula can be used, with some *reestimation* techniques, to segment the words and assign parts of speech to each word at the same time automatically.

3. Discrimination and Robustness Oriented Adaptive Learning

There are several technical problems with a general probabilistic model. First, the *model* might not be good enough to formulate the characteristics of the task under consideration. This problem can usually be relieved by using appropriate features and by considering more contextual information when constructing the model. Second, the parameters of the model might not be estimated correctly due to the lack of a large corpus. This problem can usually

be made less severe by using a larger database or better estimation techniques. Nevertheless, even if such *modeling* problem and the *estimation* problem could be resolved, it does not mean that the *ranks* of the estimated probabilistic measure are the same as the ranks of preference of the candidate segmentation patterns. Correct recognition, however, depends on the relative order of the ranks of the candidates.

The criteria of rank ordering and maximum likelihood are usually not equivalent, although they are highly correlated. Therefore, maximum likelihood estimation does not necessarily result in minimum error rate for data in the *training* set. For these reasons, the estimated parameters for the baseline models need to be adjusted to reflect the ranks of the candidate segmentation patterns. Hence, another (probably more) important issue is how to adjust the estimated likelihood measures so as to reflect the real ranks. We do this by adjusting the values of these probability terms based on the misjudged instances. By doing so, the set of parameters could be adjusted toward the goal of minimizing the error rate of the *training* corpus directly.

Furthermore, since statistical variations between a testing set and a training set are not taken into consideration in the baseline models, minimizing the error rate in the *training set* does not imply maximizing the recognition rate in an independent *testing set*, either. To enhance robustness, an extra step can be adopted to enlarge the difference in scores between the best scored candidate and the other candidates. This step will enhance the robustness of the model so that the performance will not be affected significantly by different text styles.

3.1 Adaptive Learning

The goal of adaptive learning is to provide a new parameter set, Λ', such that the new parameters in Λ' can provide more discrimination capability than the baseline parameter set Λ by adjusting the current parameters based on the misjudged training tokens. The basic idea is to adjust the parameters associated with the segmentation score of the correct candidate when the correct candidate is superseded by other candidates of larger scores; the adjustment will be continued until the modified score of the correct candidate is the largest among all candidates. Let y_k be the candidate whose segmentation score is the largest among all the candidates for the k-th training sentence, and let z_k be the correct candidate, then a distance measure $d_\Lambda(y_k, z_k)$ could be defined as a measure of separability between y_k and z_k. In particular, since we are concerned with the ranking order of the scores of the candidates, the *differences* of the segmentation scores could be used as the distance measure.

A larger difference between the segmentation scores for the correct candidate and the highest-scored candidate implies larger penalty of misjudgement. Thus, we can define a loss function as an increasing function of the distance, such as $\tan^{-1}(d_\Lambda/d_0)$ (Amari [1]), to indicate the penalty suffered from misjudgement.

To acquire a better parameter set, each parameter corresponding to the misjudged sentence is changed by a small amount in each iteration of learning so as to reduce the penalty of misjudgement; the amount of adjustment, say δA, will depend on the loss or penalty of misjudgement. Take the following segmentation patterns as an example:

1. 對 方姑娘 而言
   W1 W2 W3

2. 對方 姑娘 而言
   W1′ W2′ W3′

If model M1 is used, then the segmentation scores for these two patterns are determined by 5 parameters, namely, P1 = logP(W1), P2 = logP(W2), P3 = logP(W3) and P1′ = logP(W1′), P2′ = logP(W2′), P3′ = logP(W3′) (= P3, in this case), respectively. Assume that the initial values of these parameters are P1 = -1.8, P2 = -2.6, P3 = -1.7, P1′ = -1.6, P2′ = -2.3, and P3′ = -1.7, then the segmentation score of the first candidate (which is also the correct pattern) is -6.1 (= -1.8 -2.6 -1.7) and the segmentation score of the second candidate (which has the highest score) is -5.6 (= -1.6 -2.3 -1.7). Since this training sentence is misjudged, we may suffer from a loss whose penalty depends on the distance, namely the difference between the scores, (-5.6) - (-6.1) =0.5.

If the value of the loss function for this distance is 0.46, and the amount of adjustment, δA, for that amount of loss is 0.2, then we have a revised parameter set: P1 = -1.8+0.2 = -1.6, P2 = -2.6+0.2 = -2.4, P1′ = -1.6-0.2 = -1.8, P2′ = -2.3-0.2 = -2.5 and , P3 = P3′ = -1.7. Note that since P3 (P3′) happens to be adjusted in both patterns by the same amount, this parameter will not be changed after adjustment.

It is obvious that the correct candidate now has a higher score after parameter adjustment. Moreover, the parameters for the highest-scored candidate, which might be responsible for the misjudgement, are reduced after adjustment. So other misjudged sentences might also be affected by the adjustment of these parameters. If the correct candidate is still not the one with the highest score after the adjustment, the same procedure can be repeated; the

parameters of the correct candidate and the (possibly new) highest-scored candidate will be adjusted further until the correct candidate has the highest score.

Although the amount of adjustment for the various P(W)'s is shown to be the same in the current example, it may have to be weighted differently when we consider different information sources jointly. For instance, in model M6, we may use a smoothing technique to get a better estimated score by assigning different weights to the $P(w_k \mid l_{k-1})$ terms and the $P(t_k \mid t_{k-1})$ terms. Under such circumstance, the amount of adjustment for these two kinds of parameter sets will also be weighted by the same amount to account for their respective contributions.

Under appropriate conditions, it can be proved that the average amount of change in average loss will be *decreased* due to the adaptation (Amari [1]). Therefore, it is guaranteed that, by adjusting the parameters Λ of the baseline models in this manner, the discrimination power, in terms of the distances between the correct candidate and the other segmentation patterns, will be increased. Furthermore, since the amount of change in the parameters is directly proportional to the gradient of the loss function (Amari [1], Chiang [5], Su[9]), this also implies changing the parameters Λ in the direction in which the change in mean loss is the most drastic. Therefore, the speed of convergence is fast with this learning algorithm.

3.2 Robustness Enhancement

In addition to enhancing the *discrimination power* of the segmentation models, the *robustness* of the segmentation models is also an important concern. The robustness could be enhanced by increasing the "margin" of distances between the correct pattern and the other competing candidates (Su [9]). This can be done by adjusting the scores of the correct segmentation pattern and the one with the secondary highest score even after the correct segmentation pattern has been assigned the highest score. The adjustment of the parameters will stop only after the distance margin between the correct one and the candidate with the secondary highest score exceeds a given threshold. This will ensure that the correct candidate is separated from other competing candidates by at least the prescribed amount of margin. In this stage, the loss will be measured in terms of the distance between the top 2 candidates.

By enforcing a "margin" between the correct segmentation pattern and the most competitive candidate, the segmentation score will be more robust in the sense that any *statistical variations* between the *training corpus* and the *real instances* in the various applications could be properly suppressed. It is very important to enhance the robustness of the models in this

way, because the instances in real applications could not be predicted in advance. For more technical information on the robust adaptive learning algorithm, please refer to (Amari [1], Chiang [5], Su [9]).

4. Resolution of the Unknown Word Problem

Most word segmentation models in the literature are based on a simple assumption that all words in the text could be found in the system dictionary; there are no "unknown words" to the dictionary. However, as will be seen in a later section, such an assumption is usually unrealistic; the error introduced by unknown words, such as unknown proper nouns, constitutes a large fraction of the error rate in word segmentation. Therefore, it is important to take the unknown word problem seriously in dealing with real applications.

A word may become unknown to the system simply because it was not stored in the dictionary or because it belongs to some particular types of words, such as proper nouns, that can not be enumerated exhaustively. Sometimes, a substring of an unknown word is a legal word in the dictionary. In this case, the unknown word will be divided into pieces in the dictionary lookup process. It is also possible that an unknown word is a substring of some legal words in the dictionary. In this case, the unknown word will be hidden behind the legal word. All these error transformations: missing entry, separation of the unknown word into pieces, and hidden by a legal word, make it impossible to find all segmentation patterns by a simple dictionary lookup process.

The general solution is to take possible inverse error transformations in the vicinity of an unknown word; then evaluate the segmentation score or a revised version of it to select the most possible segmentation pattern, with unknown words recognized as a particular class of character stream of unknown length. This means to extend the segmentation patterns acquired from simple dictionary lookup by combining or dividing characters in a prescribed window where an unknown word is suspected to occur, and choose the most likely segmentation pattern from the set of extended segmentation patterns, including those candidates that are introduced by the unknown word problem. The general solution could be very complicated and will be addressed in other papers. Here, we just show a simplified version, and reveal some technical issues in unknown word resolution.

In particular, we could regard an unknown word, say w_u, as a unit of unknown length l_u that could possibly appear anywhere in the region where an unknown word is suspected to occur. We then use the dependency of the class of unknown words with their context to

determine the preference of the various segmentation patterns. The main task is to determine the positions and lengths of the unknown words in the suspected "unknown word regions" as shown below.

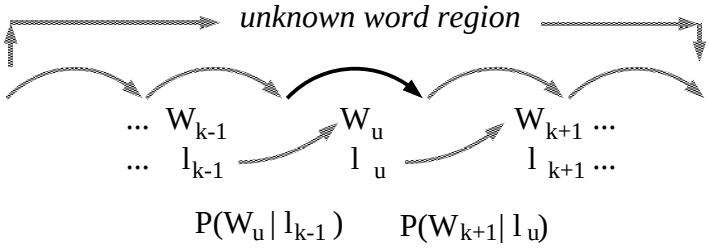

Figure 1 Evaluating segmentation score when unknown words are encountered.

For simplicity, assume that an unknown word region has been identified and exactly *one* unknown word is within the region, then we can formulate the segmentation score as in any of the previously mentioned models by replacing $w_{i,k}$ in one of the probability terms with w_u, and evaluate the segmentation scores for the various possible locations and lengths in the same way as if it was a known word. For example, if model M4 is applied to the suspected unknown word position and word length in Figure 1, we will have probability terms like:

$$score \approx \cdots \times P(w_u \mid l_{k-1}) \times P(w_{k+1} \mid l_u) \times \cdots \tag{4.1}$$

where $P(w_u \mid l_{k-1})$ is the probability that an unknown word will follow a word of length l_{k-1}, and $P(w_{k+1} \mid l_u)$ is the probability that the next word w_{k+1} will appear after an unknown word of length l_u.

The transition probabilities concerning the unknown words could be estimated from the training corpus by counting the relative frequencies of the lexical entries that could not be found in the system dictionary and the word lengths of their surrounding words.

Also, to rate the possibility that the suspected unknown word region does contain an unknown word, the above formulation must contain a factor of the form:

$$P\left(c_i^j \text{ contains an unknown word of length } l_u \text{ at position } k \mid c_1^n\right) \tag{4.2}$$

which serves to detect the unknown word regions. The detection of the unknown word regions is a nontrivial task. For the present, we just use the available word length information and the following simplified formula to account for the above factor:

$$P\left(L_{uwr}\right) \times P\left(w_u \in c_i^{i+L_{uwr}-1} \mid L_{uwr}\right) \times P\left(l_u \mid w_u \in c_i^{i+L_{uwr}-1}\right) \quad (4.3)$$

where $P\left(L_{uwr}\right)$ is the prior probability that the unknown word region ("uwr") consists of isolated single characters of length L_{uwr}; $w_u \in c_i^{i+L_{unr}-1}$ stands for the event that an unknown word does exist in the unknown word region, and $P\left(l_u \mid w_u \in c_i^{i+L_{unr}-1}\right)$ is the probability that the unknown word length in such an unknown word region is of length l_u. The results will be investigated in the analysis section.

5. Test and Analysis

5.1 Simulation

To compare the performance of the various models, a Chinese text corpus with articles from different domains is constructed for evaluation. The contents of the corpus are mostly related to politics, economics and cinema review.

The sentences are segmented by hand so that they could be used for training or testing, as well as for comparison with machine processed results. The characters between punctuation marks are segmented into smaller tokens. Because there is no common standard about the definition of Chinese words, some rules of thumb are used for manual segmentation. In particular, the following principles of segmentation are taken to keep it as consistent as possible.

1. Frequently used compound nouns and idiomatic expressions are segmented as single words without further segmentation.
2. A segment that has a direct mapping with an English word is considered a Chinese word. This technical principle is adopted specifically for the machine translation system we are working with.
3. Small segments that could be derived with general morphological rules are merged and be regarded as one word. In general, such words can be formed in the lexical analysis phase with a simple finite state machine. Therefore, the merged segments are considered a word that should be output by the segmentation algorithm as one unit.
4. When a segment is segmented into smaller tokens and the semantics of this segment can not be recovered by the compositional semantics of the smaller tokens, then the original segment will be regarded as a single word.

5. A large segment that contains a predicate part, its arguments or complements, negation markers or aspect markers is divided into smaller segments corresponding to the respective parts. This makes it easy to map each part to its syntactic or semantic construct when used for natural language applications. In fact, the purpose of word segmentation is to find the terminal words to be used by a syntactic or semantic analyzer. Therefore, those segments that could be mapped directly to the syntactic or semantic constructs are identified as such terminal words.
6. When conflicts are encountered in applying these principles, judgement is given by the human according to the frequency of use.

The testing sentences are scanned and all ambiguous segmentation patterns allowed by dictionary lookup are constructed. The various segmentation patterns are then scored with the various segmentation models. Adaptive learning as well as robustness enhancement are performed to improve the segmentation models in some testing cases. The top-1 candidate is then compared with the hand parsed results to evaluate the performance of the model under consideration.

Instead of judging the correctness by human inspection *after* the machine processed results are produced, a file is prepared to hold hand-parsed segmentations for comparison *before* the evaluation is started; the file is kept untouched throughout the evaluation process for all models. Such arrangement ensures that the evaluation is not affected by personal judgement, which may vary from one time to another, and keeps a consistent criterion of correctness.

The dictionary contains 99,441 entries, and about 9,755 words are actually encountered in the corpus. The tag set for models M5 – M7 contains a total of 22 parts of speech for Chinese and 3 special tags. (The testing environment is shown in Table 7.) To see the effects of unknown words on the performance of word segmentation, some tests are conducted in two modes, one with unknown words in the testing sentences and the other with all unknown words inserted to the dictionary.

5.2 Performance Evaluation

Since most models exhibit high recognition accuracy, the error rate, defined as “100%-Accuracy” is emphasized in performance evaluation. (The word accuracy or sentence accuracy are shown in the parentheses for comparison with other reports though.) The word accuracy is defined as the number of correctly segmented words divided by the total number of words in manually segmented sentences. The sentence accuracy, on the other

hand, is defined as the number of correctly segmented sentences divided by the total number of sentences involved in testing. Here, a sentence actually refers to a segment between the punctuation marks. A sentence is said to be "correctly segmented" if none of the words in the sentence is incorrectly identified.

Baseline Performance

Table 1 and Table 2 show the baseline performance with models M1, M2, M3 and M4 as shown in Eq. (2.2.6). In Table 1, the training and testing sentences contain unknown words, which can not be found in the dictionary. In Table 2, all unknown words are entered to the dictionary as legal entries.

Model	Training Set Error (*Accuracy)		Testing Set Error (*Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
Max Match-1	4.01 (95.99)	20.74 (79.26)	4.23 (95.77)	20.68 (79.32)
Max Match-2	4.01 (95.99)	20.77 (79.23)	4.15 (95.85)	20.54 (79.46)
P(Lk|Lk-1)	8.70 (91.30)	45.54 (54.46)	9.41 (90.59)	47.86 (52.14)
P(m|n)	7.19 (92.81)	38.61 (61.39)	7.82 (92.18)	39.30 (60.70)
P(Wk)	3.62 (96.38)	19.81 (80.19)	3.94 (96.06)	19.97 (80.03)
P(Wk|Lk-1)	3.68 (96.32)	20.08 (79.92)	4.07 (95.93)	21.04 (78.96)
(*) The numbers in the parentheses show the accuracy rates				

Table 1 Baseline Performance WITH Unknown Words

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
Max Match-1	1.14 (98.86)	4.05 (95.95)	1.22 (98.78)	4.07 (95.93)
Max Match-2	1.14 (98.86)	4.07 (95.93)	1.12 (98.88)	3.78 (96.22)
P(Lk|Lk-1)	6.16 (93.84)	37.57 (62.43)	6.82 (93.18)	40.09 (59.91)
P(m|n)	5.24 (94.76)	28.53 (71.47)	5.71 (94.29)	29.60 (70.40)
P(Wk)	0.54 (99.46)	2.07 (97.93)	0.76 (99.24)	2.50 (97.50)
P(Wk|Lk-1)	0.47 (99.53)	1.77 (98.23)	0.73 (99.27)	2.50 (97.50)

Table 2 Baseline Performance WITHOUT Unknown Words

A commonly used heuristic approach, designated as "Max(imum) Match-1", is also shown for comparison. It scans the input from left to right and from right to left, respectively, to match against the dictionary entries; the one with a smaller number of words is considered the preferred segmentation pattern. During the scanning process, if two matches against the dictionary entries are possible from the current word boundary, then the one with a larger number of characters is selected as the correct match. If the total number of words in both scanning directions are the same, then the first distinct word, either from left or from right, is compared. The segmentation pattern corresponding to the word with a larger number of characters is selected as the preferred pattern. A variant of the maximum match approach, designated as Max Match-2, as proposed in Chen [4] (Heuristic rule #1), is also implemented for comparison. It scans the text left-to-right and uses a 3–word sequence, instead of a single word, to judge the preference of the first word in this sequence.

There are several interesting and important points to point out concerning the above performance. First, it is surprising that a "trivial" model like model M2 ($P_i\left(l_k \mid l_{k-1}\right)$) or model M3 ($P\left(m_i \mid n\right)$), which uses only the word length, word count and character count information, achieve comparable performance in word accuracy as the other models that make use of word information.

As noted previously, Chinese words are mostly double-character words, single-character words and tri-character words. This implies that there might be useful information in the dependencies between word lengths and even character counts or word counts. Therefore, it is significant to use such features for segmentation. As can be seen from the tables, such a trivial model is not significantly worse than other more "reasonable" models. This means that word segmentation could be easily resolved statistically even with a simple model like model M2 or M3. Because the number of parameters for these two models are very small and the parameters do not refer to any lexical entries, they could be used in some applications where a large dictionary is unavailable.

Second, the unknown words introduce significant error rates. The word accuracy is degraded by about 2–3% in both training set or testing set, and the sentence accuracy is degraded by about 8%-19%. This means that the unknown word problem is a major source of errors for the word segmentation problem. The degradation is also observed between Table 3 and Table 4 even after adaptive learning is applied; in this case, the degradation in word accuracy is about 3% and the degradation in sentence accuracy is about 17–19%.

In Table 1, M1 model is slightly better than M4 model; in Table 2, M4 is slightly better

than M1. However, the difference in word accuracy is not more than 0.1% and the sentence accuracy differs by less than 1.1%. So it is hardly distinguishable. The same is true when we compare the corresponding rows in Table 3 and Table 4 where adaptive learning is applied. A larger difference is observed only when the tag transition probabilities ($P(t_k \mid t_{k-1})$) is jointly considered for segmentation as shown in Table 5. In general, the M4 model is slightly better than M1. Yet, both models are better with respect to the maximum match heuristics.

Adaptive Learning

Table 3 and Table 4 show the performance after the robust adaptive learning algorithm is applied to the baseline models. Since the maximum match algorithms use a deterministic process, they do not have the capability of learning. Hence, there is no corresponding entry in the tables.

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
P(Lk|Lk-1)	4.17 (95.83)	21.33 (78.67)	4.37 (95.63)	21.33 (78.67)
P(m|n)	4.33 (95.67)	22.18 (77.82)	4.43 (95.57)	21.47 (78.53)
P(Wk)	3.28 (96.72)	18.79 (81.21)	3.84 (96.16)	20.26 (79.74)
P(Wk|Lk-1)	3.23 (96.77)	18.28 (81.72)	4.00 (96.00)	21.04 (78.96)

Table 3 Performance WITH Unknown Words after LEARNING

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
P(Lk|Lk-1)	1.20 (98.80)	4.65 (95.35)	1.19 (98.81)	4.14 (95.86)
P(m|n)	1.26 (98.74)	4.99 (95.01)	1.23 (98.77)	4.21 (95.79)
P(Wk)	0.38 (99.62)	1.60 (98.40)	0.68 (99.32)	2.50 (97.50)
P(Wk|Lk-1)	0.11 (99.89)	0.48 (99.52)	0.61 (99.39)	2.35 (97.65)

Table 4 Performance WITHOUT Unknown Words after LEARNING

When comparing Table 3 and Table 4 with Table 1 and Table 2 respectively, some facts are observed. First the simple models M2 and M3 are greatly improved both in word accuracy and sentence accuracy by adaptive learning. The improved performance is comparable with the other models which use word information. The improvement for M1 and M4 models are

less obvious because the baseline performance is already very high before learning. In fact, one instance in Table 3 shows a little degradation in sentence accuracy due to over-tuning of the parameters. However, substantial error rate reduction can be observed in the other cases.

The above results confirm the underlying principle of adaptive learning that finding the correct ranks among the estimated scores, rather than finding a better estimate of the scores, plays an important role in statistical word segmentation (and virtually in all such statistical frameworks.) This may also imply that the initial baseline model might not be as important as the learning process, although it is important to have a good initial guess. Indeed, the criterion of the initial baseline models is to minimize the risk of misjudgement by maximizing the estimated probability measure. On the other hand, the robust adaptive learning algorithm try to find a direct mapping between the scores and the ranks of the candidates and try to overcome statistical variations between the training and testing sentences by minimizing the system error rate directly. Therefore, as observed in the tables, it is more robust for unseen text after learning.

Segmentation with Lexical Tags

Table 5 shows the performance when lexical tags (i.e., parts of speech) are used in word segmentation. These rows correspond to the models M5, M6, M7 in Eqn. (2.2.9). In comparison with Table 2, the baseline performance of model M5 ($P(t_k \mid t_{k-1})$), which uses lexical tags for segmentation, does not show more promising performance than M1 or M4, although its word accuracy can achieve as high as 97%. The model M1 ($P(w_k)$), when jointly considered with the lexical tag transition probability ($P(w_k) \times P(t_k \mid t_{k-1})$), is in fact degraded slightly. The baseline performance of M6 ($P(w_k \mid l_{k-1}) \times P(t_k \mid t_{k-1})$) is only slightly better than that of M4, where the tag transition probability is not used. The surprising results might be due to the very free linear order of the Chinese language.

Nevertheless, the overall performance of model M6 is the best among all when robust adaptive learning is applied. Word accuracy in this operation mode can achieve as high as 99.91% for the training set and 99.39% for the testing set. The sentence accuracy is 99.55% and 97.65% for the training set and the testing set, respectively. Since this model is to optimize the segmentation pattern and the tag sequence, it is useful for automatic tagging of plain Chinese text.

If adaptive learning is not applied to M6, its performance becomes slightly less satisfactory. Under this condition, the M4 model with adaptive learning has the best performance

among all interesting models. Since the same corpora for the M1 model could be used to acquire the required parameters $P(w_k \mid l_{k-1})$, the performance is achieved without extra cost beyond what is required for the context-independent word model (M1). Therefore, a good model along with robust adaptive learning could result in a cost-effective segmentation model without using extra resources.

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
P(Tk|Tk-1)	2.52 (97.48)	14.39 (85.61)	2.65 (97.35)	14.19 (85.81)
after learning =>	0.82 (99.18)	3.14 (96.86)	0.92 (99.08)	3.21 (96.79)
P(W)*P(Tk|Tk-1)	0.66 (99.34)	2.89 (97.11)	0.89 (99.11)	3.57 (96.43)
P(W|L)*P(Tk|Tk-1)	0.47 (99.53)	1.77 (98.23)	0.71 (99.29)	2.43 (97.57)
after learning =>	0.09 (99.91)	0.45 (99.55)	0.61 (99.39)	2.35 (97.65)
P(W|T)*P(Tk|Tk-1)	1.47 (98.53)	6.79 (93.21)	1.50 (98.50)	6.04 (93.94)

Table 5 Baseline Performance WITHOUT Unknown Words but WITH Lexical Tag Information

Lexical Tags *vs.* Learning

In contrast to adaptive learning, using lexical tags does not seem to help much in word segmentation. This can be verified by comparing the baseline performance of the $P(w_k) \times P(t_k \mid t_{k-1})$ and $P(w_k \mid l_{k-1}) \times P(t_k \mid t_{k-1})$ models in Table 5 with the performance of $P(w_k)$ and $P(w_k \mid l_{k-1})$ models in Table 4; the small amount of degradation might imply that adaptive learning is more effective in improving the baseline models than using the lexical tag information (unless adaptive learning is also applied.)

Unknown Word Problem

As described previously, the error rate introduced by unknown words is significant. Many models in the literature are based on the assumption that all words in the text could be found in the system dictionary. It is evident, however, that such an assumption is unrealistic from the experiment results. This may imply that more research energy should be directed toward *unknown word resolution* rather than the development of alternative baseline models. Table 6 shows the performance for unknown word resolution with the model proposed in the previous section; the underlying model is a revised version of the M4 model.

	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
before learning	38.06 (61.94)	85.04 (14.96)	39.64 (60.36)	86.38 (13.62)
after learning	1.78 (98.22)	8.35 (91.65)	3.59 (96.41)	15.26 (84.74)

Table 6 Performance for Unknown Word Resolution (Baseline and Learning for 10 iterations)

It is interesting to note that the performance of the *baseline* model is very low. This is probably a generic phenomena for all kinds of error correction problems; because the segmentation patterns are extended according to the error types, the candidate patterns are no more confined to the patterns that could be generated with dictionary lookup. Hence, the number of possible segmentation patterns increases drastically, and the performance of the baseline model tends to degrade. Another factor that accounts for the degradation in the baseline performance is the estimation error of the model parameters. Because all unknown words are regarded as a special class of words with the same statistical behavior, the estimated probabilities, such as the $P(w_u \mid l_{k-1})$ term, may not indicate the specific distribution of a specific unknown word under consideration. To resolve this problem, adaptive learning is essential. The learning results in the table show how unknown word errors can be recovered after adaptive learning is applied.

In comparison with the best baseline performance in Table 1 and the best learning results in Table 3, where unknown words are not handled, the error rates are reduced by 45–51% for words and 54–58% for sentences in the training set; in the testing set, the reduction in error rates amounts to 7–9% for words and 24–28% for sentences.

Of course, we also noted that some isolated single-character words are merged by mistake with this simplified error correction model. This may imply that the current features for detecting the unknown word region and the existence of the unknown words are not effective enough for detecting some instances of unknown word errors. If better features other than the sentence length, word count, and character count could be used, the improvement might be even more encouraging.

Cost Concern

The costs of the various models are directly related to the corpus size and the number of parameters to be estimated. Table 7 shows the testing environment, including the numbers of parameters for all models. Among the various models, model M2 and M3 have the smallest number of parameters. As shown in the above experiments, many models proposed here do not have significantly different performance in terms of accuracy on segmentation. The costs of the models are thus important in some applications. This seems to suggest that we could start with a simple baseline model and use an adaptive learning algorithm to acquire low cost yet high performance in word segmentation. It also suggests that we could use the less expensive models, for example, to bootstrap an automatic dictionary construction process from very limited available corpus resources.

Model	Number of Parameters	Model	Number of Parameters
P(Lk|Lk-1)	40	P(Tk|Tk-1)	625
P(m|n)	229	P(W)*P(Tk|Tk-1)	9,755+625
P(Wk)	9,755	P(W|L)*P(Tk|Tk-1)	14,473+625
P(Wk|Lk-1)	14,473	P(W|T)*P(Tk|Tk-1)	10,231+625
Training Set	41599 words / 5608 sentences		
Testing Set	10134 words / 1402 sentences		
Dictionary	99441 entries		
Lexical Tags	22 parts of speech & 3 special tags		
Ambiguity	8.6 candidates/sentences (both training set & testing set)		

Table 7 Testing Environment

6. Conclusion

In this paper, we have proposed a generalized word segmentation model for the Chinese word segmentation problem. We have shown how to use the various available information to resolve the segmentation problem based on the generalized model. It is shown that word segmentation can be resolved easily and inexpensively with the proposed statistical models. Word accuracy as high as 96% and sentence accuracy up to 80% can be achieved in the baseline model when there are unknown words. When there are no unknown words, the performance is about 99% for words and 97% for sentence.

In addition to the baseline models, a robust adaptive learning algorithm is proposed to enhance the performance of the baseline models so that these models could perform well even in handling unseen text. It is noticed that a good adaptive learning algorithm is critical to facilitate word segmentation. The reason is that a good robust adaptive learning algorithm could provide a scoring mechanism that directly minimizes the error rates both in the training corpus and the testing set. Therefore, it provides better discrimination power in ranking the large number of possible segmentation patterns.

We also find that the unknown words contribute a significant portion of the error rate. To be practical in real applications, the unknown word problem should therefore be taken seriously. In this paper, we have proposed an error correction mechanism for resolving the special unknown word problem. With such a mechanism, the error rates are reduced by 45–51% for words and 54–58% for sentences in the training set; in the testing set, the reduction in error rates amounts to 7–9% for words and 24–28% for sentences.

Throughout the framework, we had tried to use extra information from the least expensive features already available in a segmented corpus. By using the extra features of character count, word count and word length information, it is shown to improve the system performance with respect to the other models that do not use them. The use of such inexpensive features also make possible some applications where the available resource is limited.

Acknowledgement

We would like to express our gratitude to Shu-Jun Ke (Behavior Design Corporation) for her efforts in preparing the segmented corpus. Her work provides useful training and testing materials for verifying the various proposed models. We also would like to express our thanks to the Free China Times for making the text corpus available to us. Special thanks are given to the Behavior Design Corporation for providing the Chinese-English Electronic Dictionary to this research project.

References

[1] Amari, S., "A Theory of Adaptive Pattern Classifiers," *IEEE Trans. on Electronic Computers,* vol. EC-16, no. 3, pp. 299–307, June 1967.

[2] 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, Kenting, Taiwan, ROC, 1991.

[3] 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.

[4] Chen, K.-J., Shing-Huan Liu, "Word Identification For Mandarin Chinese Sentences," *Proceedings of COLING-92*, 14th Int. Conference on Computational Linguistics, pp. 101–107, Nantes, France, July 23–28, 1992.

[5] Chiang, T.-H., Y.-C. Lin and K.-Y. Su, "Syntactic Ambiguity Resolution Using A Discrimination and Robustness Oriented Adaptive Learning Algorithm," *Proceedings of COLING-92*, 14th Int. Conference on Computational Linguistics, pp. 352–358, Nantes, France, July 23–28, 1992.

[6] 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.

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

[8] 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.

[9] Su, K.-Y. and C.-H. Lee, "Robustness and Discrimination Oriented Speech Recognition Using Weighted HMM and Subspace Projection Approach," *Proceedings of IEEE ICASSP-91*, vol. 1, pp. 541-544, Toronto, Ontario, Canada. May 14-17, 1991.

[10] 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.