A Preliminary Study on Probabilistic Models for Chinese Abbreviations

Jing-Shin Chang
Department of Computer Science & Information Engineering
National Chi-Nan University
Puli, Nantou, Taiwan, ROC.
jshin@csie.ncnu.edu.tw

Yu-Tso Lai
Department of Computer Science & Information Engineering
National Chi-Nan University
Puli, Nantou, Taiwan, ROC.
s0321521@ncnu.edu.tw

Abstract

Chinese abbreviations are widely used in the modern Chinese texts. They are a special form of *unknown words*, including many *named entities*. This results in difficulty for correct Chinese processing. In this study, the Chinese abbreviation problem is regarded as an *error recovery* problem in which the suspect root words are the "errors" to be recovered from a set of candidates. Such a problem is mapped to an HMM-based generation model for both abbreviation identification and root word recovery, and is integrated as part of a unified word segmentation model when the input extends to a complete sentence.

Two major experiments are conducted to test the abbreviation models. In the first experiment, an attempt is made to guess the abbreviations of the root words. An accuracy rate of 72% is observed. In contrast, a second experiment is conducted to guess the root words from abbreviations. Some submodels could achieve as high as 51% accuracy with the simple HMM-based model. Some quantitative observations against heuristic abbreviation knowledge about Chinese are also observed.

1 Introduction

The modern Chinese language is a highly abbreviated one due to the mixed uses of ancient single character words as well as modern multi-character words and compound words. The abbreviated form and root form are used interchangeably everywhere in the current Chinese articles. Some news articles may contain about 20% of sentences that have suspect abbreviated words in them (Lai 2003). Since abbreviations cannot be enumerated in a dictionary, it forms a special class of *unknown words*, many of which originate from *named entities*. Many other open class words are also abbreviatable. This particular class thus introduces complication for Chinese language processing, including the fundamental *word segmentation* process (Chiang 1992, Lin 1993, Chang 1997) and many word-based applications. For instance, a keyword-based information retrieval system may requires the two forms, such as "立委" and "立法委員" ("legislators"), in order not to miss any relevant documents. The Chinese word segmentation process is also significantly degraded by the existence of unknown words (Chiang 1992), including unknown abbreviations.

There are many heuristics for Chinese abbreviations. Such heuristics, however, can easily break (Sproat 2002). Currently, only some quantitative approaches (Huang 1994a, 94b) are available in predicting the presentation of an abbreviation. Since such formulations regard the word segmentation process and abbreviation identification as two independent processes, they probably cannot optimize the identification process jointly with the *word segmentation* process, and thus may lose the useful contextual information. Some class-based segmentation models (Sun 2002, Gao 2003) well integrate the identification of some regular non-lexicalized units (such as named entities). However, the abbreviation process can be applied to almost all word forms (or classes of words). Therefore, this particular word formation process may have to be handled as a separate layer in the segmentation process.

To resolve the Chinese abbreviation problems and integrate its identification into the word segmentation process, this study proposes to regard the abbreviation problem in the word segmentation process as an "error recovery" problem in which the suspect root words are the "errors" to be recovered from a set of candidates according to some generation probability criteria. This idea implies that an HMM-based model for identifying Chinese abbreviations could be effective in either identifying the existence of an abbreviation or the recovery of the root words

from an abbreviation. We therefore start with a unified word segmentation model so that both processes can be handled at the same time, and when the input is reduced to a single abbreviated word, the model can be equally useful for recovering its root.

As a side effect of using HMM-based formulation, we expect that a large abbreviation dictionary could be derived from a large corpus or from web documents through the training process of the unified word segmentation model automatically.

Section 2 will show our HMM models and the three abbreviation problems correspond to the three basic HMM problems. Section 3 will show the experiment setup. Section 4 will examine the experiments to guess abbreviations from root or vice versa.

2 Chinese Abbreviation Models

2.1 An Error Recovery Paradigm

To resolve the abbreviation problems, first note that the most common action one would take when encountering an abbreviation is to find its candidate roots (probably from a large abbreviation dictionary if available or from an ordinary dictionary with some educated guesses), and then identify the most probable one. This process is identical to the operation of many spelling correction models, which generate the candidate corrections according to a *reversed* word formation process, then justify the best candidate.

Such an analogy indicates that we may use an HMM model (Rabiner 1993), which is good at finding the best *unseen* state sequence, for error recovery. There will be a direct map between the two paradigms if we regard the observed input character sequence as our "observation sequence", and regard the unseen word candidates as the underlying "state sequence".

Given these mappings, we will be able to use many standard processing approaches for HMM when we have to answer some interesting questions (including root word recovery). Among all interesting questions for an HMM, we have three basic questions to ask the model (Rabiner 1993), namely the output probability of an output sequence, the best underlying state sequence and the best parameters given a training corpus.

If we can ask the HMM for abbreviation the same questions, then we will also be able to answer the question on (1) what is the likelihood that a string is an abbreviation, (2) what are the best underlying root words for an input character string that contains abbreviations, and (3) how to estimate the model parameters automatically given a corpus.

The first question is related to the problem of generating an appropriate abbreviation from a root word; the second question is linked to finding the best underlying roots from an abbreviated string, and the third question have a direct link to the construction of an abbreviation dictionary automatically from a corpus. For now we will not explore this third question, but leave it to a research that would be launched in the near future.

The most interesting question to ask is, of course, the second question in the Chinese tokenization process. Therefore, we will start with a unified word segmentation model, which has the capability to handle abbreviation problem jointly with the word segmentation process.

2.2 HMM-Q2: Unified Word Segmentation Model for Abbreviation Recovery

To integrate the abbreviation process into the word segmentation model, firstly we can regard the segmentation model as finding the best underlying words $w_1^m \equiv w_1, \cdots, w_m$ (which include only base/root forms), given the surface string of characters $c_1^n \equiv c_1, \cdots, c_n$ (which may contain abbreviated forms of compound words.) The segmentation process is then equivalent to finding the best word sequence $\vec{w}*$ such that:

$$\begin{aligned}\vec{w}* &= \arg\max_{w_1^m : w_1^m \Rightarrow c_1^n} P\left(w_1^m \mid c_1^n\right) \\ &= \arg\max_{w_1^m : w_1^m \Rightarrow c_1^n} P\left(c_1^n \mid w_1^m\right) \times P\left(w_1^m\right) \\ &= \arg\max_{w_1^m : w_1^m \Rightarrow c_1^n} \prod_{\substack{i=1,m \\ w_i \Rightarrow \vec{c}_i}} P\left(\vec{c}_i \mid w_i\right) \times P\left(w_i \mid w_{i-1}\right)\end{aligned}$$

Equation 1. Unified Word Segmentation Model for Abbreviation Recovery

where $\vec{c}_i$ refers to the surface form of w_i, which could be in an abbreviated or non-abbreviated (or any transformed) form of w_i. The last equality assumes that the generation of an abbreviation is independent of context, and the language model is a word-based bigram model. Such assumptions can be adapted to different submodels for word segmentation (Chiang 1992) as appropriate. Furthermore, in many cases, the underlying word w_i will be a *compound* word consisting of other constituent words w_{ij} (e.g., "台灣 大學"). And, the probability $P\left(\vec{c}_i \mid w_i\right)$ is not always 1 or 0, since the constituents may be abbreviated

differently in different context, making the mapping of the compound ambiguous. For instance, some people may prefer to abbreviate '工業技術研究院' (Industrial Technology Research Institute; ITRI) into '工研院' (IRI) while other may prefer an abbreviation of '工技院' (ITI).

Notice that, this equation is equivalent to the formulation for an HMM (Hidden Markov Model) (Rabiner 1993) to find the best "state" sequence given the observation symbols. The parameters $P(w_i \mid w_{i-1})$ and $P(\vec{c}_i \mid w_i)$ represent the transition probability and the (word-wise) output probability of an HMM, respectively; and, the formulations for $P(w_1^m)$ and $P(c_1^n \mid w_1^m)$ are the respective "language model" of the Chinese language and the "generation model" for the abbreviated words (i.e., the "abbreviation model" in the current context). The "state" sequence in the word segmentation case is characterized by the root forms $w_1^m \equiv w_1, \cdots, w_m$, or the hidden words; and, the "observation symbols" are characterized here by $c_1^n \equiv c_1, \cdots, c_n \equiv \vec{c}_1, \cdots, \vec{c}_m$, where the surface form $\vec{c}_i \equiv c_{b(i)}^{e(i)}$ is a chunk of characters beginning at the b(i)-th character and ending at the e(i)-th character.

Such an analogy with an HMM enables us to estimate the model parameters using an unsupervised training method that is directly ported from the forward-backward or Baum-Welch re-estimation formula (Rabiner 1993) or a generic EM algorithm (Dempster 1977).

Note also that, while the above formulation is intended for finding root words in a sentence, with the help of contextual words, we can also apply the same formulation to a single abbreviated word (likely to have a compound word as its root in many cases) to find the most likely constituent words, without the help of surrounding words, but with the help of contextual constraints among its constituents.

2.2.1. Language Model

The word transition probability $P(w_i \mid w_{i-1})$ used in the language model is used to provide contextual constraints among root words. It may not be reliably estimated when the language has a large vocabulary and when the training corpus is small. To resolve this problem, we can back-off the bigram word transition probability to a unigram word probability using Katz's method (Katz 1987) for rare bigrams. We can, of course, use other smoothing methods to acquire reliable parameters. The smoothing issues, however, are not the main focus of this preliminary study.

2.2.2 Generation Model for Abbreviations

In the perfect case where all words are lexicalized, rendering all surface forms identical to their "root" forms and all words are known to the system dictionary, we will have $P(\vec{c}_i \mid w_i) = 1$, $\forall i = 1, m$, and **Equation 1** is no more than a word bigram model for word segmentation (Chiang 1992). In the presence of unknown words (e.g., abbreviations being one of such entities), however, we can no longer ignore the generation probability $P(\vec{c}_i \mid w_i)$.

For example, if $\vec{c}_i$ is '台大' then w_i could be the compound word '台灣 大學' (Taiwan University) or '台灣 大聯盟' (Taiwan Major League). In this case, the parameters in P(大學|台灣) x P(台|台灣) x P(大|大學) and P(大聯盟|台灣) x P(台|台灣) x P(大|大聯盟) will indicate how likely '台大' is an abbreviation, and which of the above two compounds is the root form of the abbreviation. Therefore, we need a method for estimating the probabilities between the abbreviations and their root forms (many of which are compound words with other constituents).

2.3 Applying Abbreviation Models

There are two problems to use the unified model which takes abbreviated words into account. First of all, since the word lattice is constructed from all possible $w_1^m \equiv w_1, \cdots, w_m$, how can we construct it without really knowing the candidate base forms of $\vec{c}_i$ in advance? We don't really want to randomly combine all possible root forms, which is not affordable in computational cost. Therefore, we have to make some smarter choices. Second, how to compute the abbreviation (output/generation) probability $P(\vec{c}_i \mid w_i)$ once the lattice is constructed with candidate root words?

2.3.1 Candidate Root Word Generation

The first problem can be resolved if we choose some highly probable constituents w that would generate each individual characters c_{ij} in $\vec{c}_i$ *independently*, and allow such Top-N candidates to form part of the complete word lattice. That is,

for each individual character c_{ij}, we choose its Top-N candidates according to: $P(c_{ij} \mid w) \bullet P(w)$.

The probability $P(c_{ij} \mid w)$ here represents the character-wise generation probability of a single character from its corresponding root word. Notice that, after we apply the word segmentation model **Equation 1** to the word lattice, some of the above candidates may be preferred and others be discarded, by consulting the neighboring words and their transition probabilities. This makes the abbreviation model *jointly* optimized in the word segmentation process, instead of being optimized independent of context.

2.3.2 Abbreviation Probability

The second problem can be resolved using the following equation if w_i can be segmented into $w_i \equiv w_{i1}, \cdots, w_{iL}$, each constituent corresponding to a character in $\vec{c}_i \equiv c_{b(i)}^{e(i)}$:

$$P(\vec{c}_i \mid w_i) = P\left(c_{b(i)}^{e(i)} \mid w_{i1}^{iL}\right)$$
$$= \prod_{j=1,L} P\left(c_{b(i)+j-1} \mid w_{ij}\right) \cdot P\left(w_{ij} \mid w_{i,j-1}\right)$$
$$L = e(i) - b(i) + 1 \equiv L(i)$$

Equation 2. Abbreviation Probability.

In other words, we use the transition probability between constituent words and the character-wise generation probabilities of individual characters from a constituent word to estimate the global generation probability of the abbreviated form.

2.3.3 Simplified Abbreviation Models

It is sometimes simply not efficient to save all pairs of root compounds and their respective abbreviations in an abbreviation dictionary. Therefore, it is desirable to simplify the abbreviation probability by using some simpler features for Chinese abbreviation words. For instance, it is known that many 4-character compound words will be abbreviated as 2-character abbreviations (such as the case for the <台灣大學, 台大> pair.) It was also known heuristically that many such 4-character words are abbreviated by reserving the first and the third characters, which can be represented by a '1010' bit pattern, where a '1' means to reserve the respective character and a '0' means to delete it. Therefore, a reasonable simplification for the abbreviation model is to introduce the *length* and the *bit pattern* for abbreviation operations as additional features into the abbreviation model. If this is the case, we will have the following augmented abbreviation model.

$$P(\vec{c} \mid w) = P(c_1^m, bit, m \mid r_1^n, n)$$
$$\equiv P(c_1^m \mid r_1^n) \times P(bit \mid n) \times P(m \mid n)$$

where
- c_1^m : surface characters.
- r_1^n : root word characters.
- m: length of surface characters.
- n: length of root word characters.
- bit: bit pattern of abbreviation

Equation 3. Abbreviation Probability using Abbreviation Pattern and Length Features.

All these three terms can be combined freely to produce as many as 7 sub-models for the abbreviation model. Note, the first term $\Pr(c_1^m \mid r_1^n)$ plays the same role as the older notation of $\Pr(\vec{c} \mid w)$, which could means a pair of <abbreviation, root> or be evaluated as the product of the per-character generation probabilities and the sub-constituent transition probabilities as outlined in **Equation 2**. This term can of course be ignored from the above augmented abbreviation model so that only very simple length and position features are used for abbreviation handling.

3 Data and Parameter Estimation

An abbreviation dictionary containing the word-abbreviation pairs is required to test the proposed models. Unfortunately, a large Chinese abbreviation dictionary is not available. Therefore, we have to collect some of the generic abbreviations, and make others manually from some named entity lists. Almost half of our collection comes from the Ministry of Education of the ROC. (http://www.edu.tw/clc/dict/). (In a future plan, a large abbreviation dictionary will be built automatically by using the proposed models.)

Eventually, we got 1547 root-abbreviation pairs. Among them, 1235 pairs are considered *simple* and 312 pairs are "*tough*" in the sense that they violate some model assumptions. For instance, we required that a root in a compound word be mapped to at least one character in its abbreviation (not to a null string), and we also assume that the word cannot be mapped to a character that is not part of the word. (For example, AB can be abbreviated as A or B but not C.) Some tough words will actually map substrings to *null* strings;

others may be *recursively* abbreviated; and yet others may change the word order (as in abbreviating "第一核能發電廠" as "核一廠" instead of "一核廠".). As a result, the tough pairs will not be handled correctly with current models.

To simplify the task, only the 1235 simple pairs are tested for evaluation. They are further divided randomly into a training set of 986 pairs (80%) and a test set of 249 pairs (20%). Since the corpus size is not large, the compound words are also manually segmented into their constituents in order to know the true alignments between each character of the abbreviation with its root form in the compound word. Admittedly, such an extremely small training set causes serious data sparseness problem during training. Therefore, the evaluated performance in this preliminary report will be highly underestimated.

The parameters are estimated in the *unsupervised* mode using a standard EM algorithm or the re-estimation method as conventional HMM models would do (Rabiner 1993). In addition, the manually segmented dictionary also allows us to estimate the model parameters in the *supervised* mode.

The unsupervised training will automatically align each character in the abbreviations to its root form in the full words. It is observed that 65.5% of the training set dictionary pairs will be aligned correctly. Other pairs are aligned partially correct.

Note that parameters $P(m|n)$ and $P(bit|n)$ can be estimated using maximum likelihood estimation by directly consulting the abbreviation dictionary since they are only related to word length and character position. It is interesting, in the first place, to check these types of parameters quantitatively to see if they reveal some abbreviation heuristics recognized by native Chinese speakers. The high frequency patterns, which are much more frequent than the ones ranked in lower places, are listed in **Table 1** and **Table 2**.

P(m|n)	Score	Examples
P(1|2)	1.00	(港|香港), (適|適用)
P(2|3)	0.67	(投縣|南投縣), (交部|交通部)
P(2|4)	0.95	(歐盟|歐洲聯盟), (暨大|暨南大學)
P(3|5)	0.73	(中研院|中央研究院), (資工系|資訊工程系)
P(3|6)	0.70	(農工系|農業工程學系), (雲科大|雲林科技大學)
P(3|7)	0.76	(廣電處|廣播電視事業處), (中科院|中山科學研究院)

Table 1. High Frequency Abbreviation Patterns [by lengths]

P(bit|n)	Score	Examples
P(10|2)	0.87	(德|德國),(美|美國)
P(101|3)	0.44	(宜縣|宜蘭縣), (限級|限制級)
P(1010|4)	0.56	(公投|公民投票), (清大|清華大學)
P(10101|5)	0.66	(環保署|環境保護署), (航警局|航空警察局)
P(101001|6)	0.51	(化工系|化學工程學系), (工工系|工業工程學系)
P(1010001|7)	0.55	(國科會|國家科學委員會), (中科院|中山科學研究院)
P(10101010|8)	0.21	(一中一台|一個中國一個台灣), (一大一小|一個大人一個小孩)

Table 2. High Frequency Abbreviation Patterns [by P(bit|n)]

Table 1 shows how word lengths will change during the abbreviation process, and **Table 2** shows which characters will be deleted from the root of a particular length. The tables quantitatively support some general heuristics for native Chinese speaker. For instance, most words will be abbreviated by deleting about half the characters in the words, as shown in **Table 1**. The data also shows that the first character in a two-character word will be retained in most cases, and the first and the third characters in a 4-character word will be retained in 56% of the cases. However, the tables also shows that around 50% of the cases cannot be uniquely determined simply by consulting the word length for its abbreviated form. This does suggest the necessity of an abbreviation model for resolving this kind of unknown words and named entities.

4 Experiments and Analysis

The unified model can be applied to a whole sentence which contains abbreviations during word segmentation. When the input is reduced to a single abbreviated word (or compound), it can also be applied to recover the underlying root constituent words (without consulting contextual words). In this paper, we will only focus on the abbreviation word recovery problems.

Two major experiments are conducted. The first experiment is to guess the most likely abbreviation form for a word using various feature combinations; the second is to guess the root word

from an abbreviation. The following sections will give more details.

4.1 Guessing Abbreviations from Roots

The main task of this experiment is to guess the most probable abbreviation forms for the unabbreviated words in a word list. The abbreviation forms of a word can be enumerated by arbitrarily retaining some characters of this root word and deleting others. For example, the word "南投縣" has six possible abbreviated forms: "南", "投", "縣", "南投", "南縣" and "投縣". In general, if we have a root word of length ***L***, there could be $2^L - 2$ possible abbreviations for this root word (excluding the word itself and the null string).

The best possible abbreviation form $\vec{c}^*$ for an input word w_i can be determined as the one with the highest generation probability $P(\vec{c}_i \mid w_i)$, i.e., $\vec{c}^* = \arg\max_{\vec{c}_i} P(\vec{c}_i \mid w)$. The generation probability for a candidate $\vec{c}_i$, in turn, can be estimated by summing up all probabilities of alignments between each character c_{ij} in $\vec{c}_i$ and the suspect constituent words w_{ij} in w_i. In other words, we have

$$P(\vec{c}_i \mid w_i) = \sum_{A \in \text{ all alignments}} P(\vec{c}_i, \text{A} \mid w_i) \equiv \sum_A P_A(\vec{c}_i \mid w_i)$$

where $P_A(\vec{c}_i \mid w_i)$ is the generation probability for a known alignment *A*, which can be estimated as in **Equation** 2.

For simplicity, we assume that each character in $\vec{c}_i$ will be mapped to a substring w_{ij} in w_i. In other words, we assume that the mapping between the constituents is 1-1, and no 1-0 or 0-1 mapping is possible. (In future works, such a constraint could be removed.) Also, we will assume that w_{ij} should at least contain the character that is aligned to it. (This is not always true for Chinese abbreviations. For example, "福建" can be abbreviated with its ancient location name "閩", which does not appear anywhere in its root.)

There is also a *normalization* issue in computing the probability of a particular alignment. In general, a shorter string may be preferred as the best abbreviation simply because it multiplies less probability factors when estimating the alignment probability. To reduce this effect, we intentionally scale down, by a normalization factor, the generation probabilities for those alignments that map a complete word into a single character. In fact, there are only about 10% of such alignments, and many of which are mapping a two-character word into a single character (which can be compensated by the large Pr(1|2) factor in the model. This simple normalization approach actually improves the test set performance greatly.

The following table shows the test set performance for using different features in the abbreviation probability as given in **Equation 3**. (The training set performance ranges from 94% to 98%, which suggests a good fit to the training data.)

Feature	Unsupervised				Supervised			
P(c|w)xP(wi|wi-1)	1	1	1	1	1	1	1	1
P(bit|n)	1	1	0	0	1	1	0	0
P(m|n)	1	0	1	0	1	0	1	0
Accuracy Rate(%)	68	68	61	60	72	70	61	58

Table 3. Test Set Performance for Abbreviation Generation with Combined Features.

Each column shows the test set performance for a submodel, which is identified by the features used for estimating the probability. The label '1' (or '0') indicates that the feature at the first column is used (or unused) in the submodel. For instance, the submodel of the second column ('111') uses all the features, including the word transition probability, word-to-abbreviation probability, probability for mapping n character word to a particular abbreviation bit pattern P(bit|n), and the probability for mapping n-character words into m-character abbreviations.

It is seen that supervised training acquires a little better performance than its unsupervised (EM) counterpart. Although not shown in this table, it is observed that the word transition probability and word-to-abbreviation probability in general should be used to get better performance. The table also shows that the other two features based on character positions and word lengths provide additional help. In particular, P(bit|n) seems to be more helpful than P(m|n) since it contains detailed information for retaining characters at particular positions.

The best performance is about 72% when supervised training is used and all the three types of features are used for estimating the abbreviation probability.

4.2 Guessing Roots from Abbreviations

In this experiment, we are given an abbreviation list; the goal is to guess the best root words of the abbreviations in the list. The parameters used here are acquired from human tagged alignments in a supervised manner.

To find the best root candidates of an abbreviated compound word, we need to find the candidate root words for each input character first. The candidate root words can be found from the training set whose generation probability $P(\vec{c}_i \mid w_i)$ is non-zero. The Top-N candidates can then be picked up as described earlier.

For instance, if we want to find the root words of the abbreviation "立委", and the probabilities P(立|立法) and P(委|委員) are non-zero, then we have the chance to recover the abbreviation "立委" back to the correct compound word "立法委員" , which consists of the candidate root words "立法" and "委員" for the input characters "立" and "委" respectively.

Unfortunately, the limited abbreviation dictionary we have is highly sparse. Among the 249 abbreviations in the test set, only 144 (58%) of them have their candidate root words available in the training set. The other 105 abbreviations (42%) cannot be recovered since each of them has at least one character whose candidate cannot be discovered from the training set. For this reason, we will limit ourselves to the performance of the "trainable" test set consisting of the 144 abbreviations, in order to factor out the sparseness problem pertaining to the training corpus.

Under such a restricted environment, we have tested various submodels to see how different language models and simple smoothing affect the results of this error recovery process. The results are summarized in the following table:

LM	SM?	Top-N	TR(%)	TS(%)	Best?
bigram	No	all	90.9	35	
		2	69.6	43	2
		1	46.0	45	1
unigram	No	all	44.2	44	
bigram	Yes	all	90.7	51	1
		2	69.6	51	1
		1	46.0	45	

Table 4. Abbreviation Recovery Performance.

Notations: LM: Language Model, SM?: Apply Smoothing?, Top-N: maximum number of Top-N candidate root words for each character, TR: Training Set Accuracy Rate, TS: Test Set Accuracy, Best?: Best TS Performance among all N's? (1 = yes, 2= rank 2)

The bigram language model uses $P(w_i \mid w_{i-1})$ in the unified HMM model while the unigram model uses $P(w_i)$ instead. Both of them use maximum likelihood estimation over the manually tagged abbreviation-root pairs when smoothing is not applied. When smoothing is applied, the smoothed bigram probability is acquired by linearly interpolating the unigram and bigram probabilities with an equal weight (0.5). The above table indicates that using the less complicated unigram model generally improve the test set performance significantly (from 35% to 44%). If the model parameters are smoothed, the improvement is even greater. Such results can be well expected in the current environment where the training data is very sparse.

Overall, the best test set performance is about 51% when using a smoothed bigram language model; and this can be achieved by using at most 2 Top-N candidate root words while constructing the underlying word lattice. This suggests that we don't really need to wildly enumerate all possible candidate root words for each input character with this model.

5. Concluding Remarks

Chinese abbreviations, a special form of unknown words and named entities, are widely seen in the modern Chinese texts. This results in difficulty for correct Chinese processing. In this preliminary study, the Chinese abbreviation problem is modeled as an *error recovery* problem in which the suspect root words are to be recovered from a set of candidates. An HMM-based model is thus used for Chinese in either abbreviation identification, or in the recovery of the root words from an abbreviation. By extending a simple abbreviation string into a whole text involving abbreviations, it can also be applied to the Chinese word segmentation for identifying abbreviations in a text, or for bootstrapping an abbreviation dictionary from a text corpus.

With the proposed model, the abbreviated forms can be guessed from root words at about 72% correction. The recovery of the root words from abbreviations is conducted at about 51% accuracy rate. Although further improvement is possible, the preliminary results are encouraging. In the near future, *bootstrapping* a large abbreviation dictionary from web text by applying the proposed models is planned. This should partially resolve the data sparseness problems. Such models will also be integrated into a Chinese

word segmentation model to partially resolve the unknown word and named entity identification problems in the tokenization process. It is expected that more applications will rely on such models for Chinese processing.

References

Chang, Jing-Shin and Keh-Yih Su, 1997. "An Unsupervised Iterative Method for Chinese New Lexicon Extraction", *International Journal of Computational Linguistics and Chinese Language Processing (CLCLP)*, 2(2): 97-148.

Chiang, Tung-Hui, Jing-Shin Chang, Ming-Yu Lin and Keh-Yih Su, 1992. "Statistical Models for Word Segmentation and Unknown Word Resolution," *Proceedings of ROCLING-V*, pages 123-146, Taipei, Taiwan, ROC.

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

Gao, Jianfeng, Mu Li, Chang-Ning Huang, 2003. "Improved Source-Channel Models for Chinese Word Segmentation," *Proc. ACL 2003*, pages 272-279.

Huang, Chu-Ren, Kathleen Ahrens, and Keh-Jiann Chen, 1994a. "A data-driven approach to psychological reality of the mental lexicon: Two studies on Chinese corpus linguistics." In *Language and its Psychobiological Bases*, Taipei.

Huang, Chu-Ren, Wei-Mei Hong, and Keh-Jiann Chen, 1994b. "Suoxie: An information based lexical rule of abbreviation." In *Proceedings of the Second Pacific Asia Conference on Formal and Computational Linguistics II*, pages 49–52, Japan.

Katz, Slava M., 1987. "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. ASSP-35 (3)*.

Lai, Yu-Tso, 2003. *A Probabilistic Model for Chinese Abbreviations,* Master Thesis, National Chi-Nan University, ROC.

Lin, Ming-Yu, Tung-Hui Chiang and Keh-Yih Su, 1993. "A Preliminary Study on Unknown Word Problem in Chinese Word Segmentation," *Proceedings of ROCLING VI*, pages 119-142.

Rabiner, L., and B.-H., Juang, 1993. *Fundamentals of Speech Recognition*, Prentice-Hall.

Sun, Jian, Jianfeng Gao, Lei Zhang, Ming Zhou and Chang-Ning Huang, 2002. "Chinese named entity identification using class-based language model," *Proc. of COLING 2002*, Taipei, ROC.

Sproat, Richard, 2002. "Corpus-Based Methods in Chinese Morphology", *Pre-conference Tutorials*, COLING-2002, Taipei, Taiwan, ROC.