Computational Linguistics and Chinese Language Processing
Vol. 12, No. 3, September 2007, pp. 325-348

Exploiting Pinyin Constraints in Pinyin-to-Character Conversion Task: a Class-Based Maximum Entropy Markov Model Approach

Jinghui Xiao*, Bingquan Liu*, and Xiaolong Wang*

Abstract

The Pinyin-to-Character Conversion task is the core process of the Chinese pinyin-based input method. Statistical language model techniques, especially ngram-based models, are mostly adopted to solve that task. However, the ngram model only focuses on the constraints between characters, ignoring the pinyin constraints in the input pinyin sequence. This paper improves the performance of the Pinyin-to-Character Conversion system through exploitation of the pinyin constraints. The MEMM framework is used to describe the pinyin constraints and the character constraints. A Class-based MEMM (C-MEMM) model is proposed to address the MEMM efficiency problem in the Pinyin-to-Character Conversion task. The C-MEMM probability functions are strictly deduced and well formulized according to the Bayes rule and the Markov property. Both the cases of hard class and soft class are well discussed. In the experiments, C-MEMM outperforms the traditional ngram model significantly by exploitation of the pinyin constraints in the Pinyin-to-Character Conversion task. In addition, C-MEMM can well utilize the syntax and semantic information in word class and further improve the system performance.

Keywords: Pinyin-to-Character Conversion, MEMM, Class-Based

1. Introduction

The standard keyboard was initially designed for native English speakers. In Asia, such as China, Japan and Thailand, people cannot input their language through the standard keyboard directly. Asian text input becomes one of the challenges for computer users in Asia. Therefore, an Asian language input method is one of the most difficult problems in Asian language processing.

* School of Computer Science and Techniques, Harbin Institute of Technology, Harbin, 150001, China
E-mail: {xiaojinghui, liubq, wangxl}@insun.hit.edu.cn

[Received October 12, 2006; Revised May 15, 2007; Accepted May 23, 2007]

For Chinese, the input methods can be roughly divided into two types: one is the structure-based or shape-based input method, which was developed based on the structure of Chinese characters, such as the Wubi method [Wang 2005], Cangjie method, Boshiamy method, among others. These methods can reach a high input speed by a skilled user. However, a lot of effort is required to master them. The other is the pronunciation-based input method, such as the Insun input method [Wang 1993], Microsoft input method, Bopomofo, among others. These methods are easy to learn. The user can input the Chinese character with scarcely any training, on the condition that they can pronounce it correctly. Hybrid input methods have also been proposed, *i.e*. Renzhi and Tze-loi input method. However, they only possess a limited share of the market.

The Pinyin-based input method is one of the most important pronunciation-based input methods. Pinyin is a system of Romanization for standard Mandarin. It uses Roman letters to represent the sound of Chinese characters. Hanyu Pinyin is the most common variant of pinyin in use. It was approved in 1958, and the government of the People's Republic of China has adopted Hanyu pinyin as the phonetic instruction in the mainland of China. In 1979, Hanyu pinyin was adopted by the International Organization for Standardization (ISO) as the standard Romanization for modern Chinese [ISO 7098: 1991]. The Pinyin-based input method dominates the market of Chinese input methods. It is said that over 97% of Chinese computer users are using pinyin to input Chinese [Chen 1997].

According to the scale of input unit, the pinyin-based input method can be divided into three types: the character-level input method, the word-level or phrase-level input method, and the sentence-level input method, respectively. The sentence-level input method usually achieves higher accuracy by exploitation of more context information than the other two. It has become the most prevalent pinyin-based input method. The Pinyin-to-Character Conversion task aims to convert a sequence of pinyin strings into one Chinese sentence. It is the core process of the sentence-level pinyin-based input method. Therefore, improving the performance of the Pinyin-to-Character Conversion system is well worth studying. In addition, the Pinyin-to-Character Conversion task can be taken as a simplified task of automatic speech recognition, both of which aim to convert phonetic information into character sequence. However, the Pinyin-to-Character Conversion task doesn't have to deal with acoustic ambiguity because the pinyin strings are directly input through the keyboard. Therefore, the technique is also illuminative in the task of automatic speech recognition.

The linguist approach [Wang 1993; Hsu and Chen 1993; Kuo 1995] and the statistical approach [Zhang *et al*. 1998; Xu *et al*. 2000; Wu 2000; Gao *et al*. 2002; Gao *et al*. 2005; Xiao *et al*. 2005a] are two technical approaches to the Pinyin-to-Character Conversion task. The statistical approach is mainly based on the technique of statistical language models, especially the ngram model and its variant forms. In recent years, it has drawn great interest due to its

efficiency and robustness. However, several drawbacks have also been found in the traditional ngram model. First, according to Zipf's law [Zipf 1935], there are a lot of words which rarely or never occur in the training corpus. The data sparseness problem is severe [Brown *et al*. 1992] in the ngram model. Second, long distance constraints are difficult to capture since the ngram model only focuses on local lexical constraints. Third, it's hard to utilize the linguistic knowledge of the ngram model.

Many techniques have been proposed to address the drawbacks of the traditional ngram model. To solve the data sparseness problem, various kinds of smoothing techniques have been proposed, such as additive smoothing [Jeffreys 1948], Katz smoothing [Katz 1987], linear interpolation smoothing [Jelinek and Mercer 1980], semantic based smoothing [Xiao *et al*. 2005b; Xiao *et al*. 2006]. To utilize the linguistic knowledge, a set of linguistic rules are generated automatically and they are incorporated into the traditional ngram model by a hybrid ngram model [Wang *et al*. 2005]. Hsu [Hsu 1995] proposes the context sensitive model (CSM) in which the semantic patterns are captured by the templates. As much as 96% accuracy, which is the best result of the traditional Chinese input methods as far as we know, is reported for CSM on the Phoneme-to-Character Conversion task. Trigger techniques have been proposed [Zhou and Lua 1998] and word-pair techniques have been proposed [Tsai and Hsu 2002; Tsai *et al*. 2004; Tsai 2005; Tsai 2006]. The linguist knowledge can be effectively described by triggers and pairs; meanwhile, the long distance constraints can be well captured. Compared with the commercial input system (MS-IME 2003), effective improvements have been achieved by these techniques [Tsai 2006]. Wang [Wang *et al*. 2004] utilizes the theory of rough set so as to discover the linguistic knowledge and incorporate it into the Pinyin-to-Character Conversion system. Compared with the traditional ngram model, Wang's system achieves a higher accuracy with a smaller storage requirement. Xiao [Xiao *et al*. 2005a] incorporates the word positional information into the Pinyin-to-Character Conversion system and achieves encouraging results in experiments. Gao [Gao *et al*. 2005] proposes the Minimum Sample Risk (MSR) principle to estimate the parameters of the ngram model. Success has been achieved with this principle for a Japanese input method.

What's more, some techniques have been proposed especially for Chinese text input method. A Pinyin-to-Character Conversion system with spelling-error correction was developed by Zhang [Zhang *et al*. 1997]. In the system, a rule-based model is designed to correct typing errors when the user inputs pinyin strings. Not only can the system accept the correct pinyin input, but it can also tolerate common typing errors. Similar work has been done by Chen [Chen and Lee 2000]. Chen constructs a statistical model to correct user typing errors. Moreover, Chen proposes a modeless input technique in which the user can input English using a Chinese input method, not requiring language mode switch.

However, there is another drawback of the ngram model in the Pinyin-to-Character

Conversion task, which has been ignored by most researchers. It takes no account of pinyin constraints on the input pinyin sequence while actually in the process of Pinyin-to-Character Conversion. This paper regards that the pinyin information from the pinyin sequence is helpful for selecting the correct character sequence in the Pinyin-to-Character Conversion task. First, the current input pinyin string is helpful for selecting the correct character which corresponds to that pinyin. For example, the input pinyin sequence is "ta1 shi4 di2 shi4 you3?" which should be converted into "他是敌是友?" ("Is he an enemy or friend?"). Let's focus on the third pinyin string of "di2". There are two homonyms which correspond to it: "敌" and "的". (There are actually many homonyms, but let 's only focus on "敌" and "的" for simplification). "的" is one of the most frequent Chinese characters and its frequency is usually much higher than "敌". According to the ngram model, the above pinyin sequence should be converted into "他是的是友?" which is a wrong conversion. However, "的" is a polyphone which corresponds to both "di2" and "de5". In Chinese, "的" is usually pronounced as "de5" instead of "di2". ("的" is pronounced as "di2" only in the word "的确" (certainly)). The frequency of "的" mainly comes with its pronunciation "de5". If the pinyin information is considered in the above conversion, the co-occurrences of "的" and "di2" are usually lower than that of "敌" and "di2". Then, the above pinyin sequence is correctly converted into "他是敌是友?". Second, the contextual information, especially the future information, can be well exploited in the pinyin constraints. For example, there are two pinyin sequences. The first one is "yi4 zhi1 ke3 ai4 de5 xiao3 hua1" which should be converted into "一枝可爱的小花" (This is a lovely flower). The second pinyin "zhi1" should be converted into "枝" which is determined by its future character "花" (flower). The second pinyin sequence is "yi4 zhi1 ke3 ai4 de5 xiao3 hua1 mao1" which should be converted into "一只可爱的小花猫" (This is a lovely cat). The second pinyin "zhi1" should be converted into "只" which is determined by its future character "猫" (cat). However, according to the ngram model, the conversion of "zhi1" is only determined by its history information which is the character "一" in the above two cases. The characters of "花" and "猫" are both the further information that the ngram model can not exploit. Therefore, the same probabilities are assigned to both the characters of "只" and "枝". They can not be distinguished by the ngram model. In the above two conversions, at least one of them would be converted incorrectly. However, if the pinyin constraints are considered, the constraints of "hua1" and "mao1", which correspond to the characters of "花" and "猫", are exploited and imposed on the conversion of "zhi1". Then, the above two cases can be distinguished and the correct conversions can be obtained. Third, the long distance constraints can be exploited from the pinyin sequence. As for the ngram model, it has to construct a high-order model to capture the long distance constraints. However, high-order ngram models suffer from the curse of dimensionality which usually leads to a severe data sparseness problem. The current model order is usually 2 or 3. In the above example, in order to exploit

the constraints of "花" and "猫" on the conversion of "zhi1", it has to build up at least a 7-order ngram model which suffers from a great data sparseness problem and cannot work well in reality. However, the pinyin constraints are collected as features and exploited under the Maximum Entropy (ME) framework in this paper. The context window size can be relatively large (*i.e*. 5 pinyin strings or 7 pinyin strings) without the curse of dimensionality. Then the constraints of "花" and "猫" can be imposed on the conversion of "zhi1" by exploitation of their pinyin information.

This paper aims to improve the performance of the Pinyin-to-Character Conversion system by exploitation of the pinyin constraints from pinyin sequence. The pinyin constraints are described under the ME framework [Berge *et al*. 1996], and the character constraints are modeled by the traditional ngram model. Combining these two models into a unified framework, the paper builds the Pinyin-to-Character Conversion system on a MEMM model [McCallum *et al*. 2000]. However, the label set on the Pinyin-to-Character Conversion task is the Chinese lexicon. The scale of Chinese lexicon is usually in the range of $10^4 \sim 10^6$, which is too large for the current training algorithms of MEMM. Therefore MEMM cannot be directly applied to the Pinyin-to-Character Conversion task. This paper involves the addition of the class of target label into traditional MEMM and proposes a Class-based Maximum Entropy Markov Model (C-MEMM) so as to solve the MEMM efficiency problem in the Pinyin-to-Character Conversion task. In C-MEMM, the pinyin constraints are first imposed on the class sequence instead of the target label sequence as MEMM does. The classes of target label can be obtained by some automatic algorithms [Li 1998; Chen and Huang 1999; Gao *et al*. 2001] or from some pre-defined thesauri [Mei *et al*. 1983]. The scale of class set is usually much smaller than that of target label, which makes it feasible to train C-MEMM under the Maximum Entropy principle. Then, these constraints are conveyed from the class sequence to the target label sequence. So, C-MEMM can efficiently exploit the pinyin constraints from pinyin sequence and get effective improvement in the Pinyin-to-Character Conversion task.

The paper is organized as follows: the MEMM model is briefly reviewed in Section 2. In Section 3, the C-MEMM model is proposed and its probability functions are deduced according to the Bayes rule and the Markov property. Both the cases of hard class and soft class are discussed in detail. Experimental results and discussions are provided in Section 4. The related works are described in Section 5, and the conclusions are drawn in Section 6.

2. Brief Review of MEMM

MEMM is a powerful tool used to perform the sequence labeling task, which is to determine a state sequence according to the observation sequence. Different from the ngram model, MEMM not only makes use of the constraints between states but also utilizes the constraints from observations. MEMM integrates these two kinds of constraints into a uniform

conditional probability function. More formally, given the observation sequence of $O = o_1, o_2 ... o_n$ and the state sequence of $S = s_1, s_2 ... s_n$, MEMM estimates the conditional probability of $P(S \mid O)$. The probability function of MEMM can be deduced in the following way:

$$
\begin{aligned}
P(S \mid O) &= p(s_1, s_2 ... s_n \mid o_1, o_2 ... o_n) \\
&\quad \textit{Bayesian Rule} \\
&= p(s_1 \mid o_1, o_2 ... o_n) p(s_2, s_3 ... s_n \mid o_1, o_2 ... o_n, s_1) \\
&\quad \textit{Markov Property} \\
&= p(s_1 \mid o_1) p(s_2, s_3 ... s_n \mid o_1, o_2 ... o_n, s_1) \\
&\quad \textit{Bayesian Rule} \\
&= p(s_1 \mid o_1) p(s_2 \mid o_1, o_2 ... o_n, s_1) p(s_3 ... s_n \mid o_1, o_2 ... o_n, s_1, s_2) \\
&\quad \textit{Markov Property} \\
&= p(s_1 \mid o_1) p(s_2 \mid s_1, o_2) p(s_3 ... s_n \mid o_1, o_2 ... o_n, s_1, s_2) \\
&\qquad \cdots\cdots\cdots\cdots \\
&= p(s_1 \mid o_1) \prod_{i=2}^{n} p(s_i \mid s_{i-1}, o_i)
\end{aligned}
\tag{1}
$$

MEMM estimates the probability of $p(s_i \mid s_{i-1}, o)$ under the ME principle so as to utilize the overlapping features. The ME principle assumes that the trained model should be consistent with certain constraints derived from the training data; meanwhile the model should make the fewest assumptions about the data. To predicate the current state *s*, the contextual information of *s* is extracted from the training data and represented as the feature function:

$$
f(h, s) = \begin{cases} 1 & \text{if } h = h^* \text{ and } s = s^* \\ 0 & \text{otherwise} \end{cases}
\tag{2}
$$

where *h* is the contextual information of state *s* and h^* (or s^*) is the concrete instance of *h* (or s). The following constraints are imposed so that the expectation of each feature in the learned model should be consistent with its empirical value in the training corpus. More formally, the constraints can be expressed as:

$$
E_p(f) = E_{\tilde{p}}(f)
\tag{3}
$$

where $E_p(f)$ is the model expectation and is defined as:

$$
E_p(f) = \sum_{h,s} \tilde{p}(h) p(s \mid h) f(h, s)
\tag{4}
$$

and $E_{\tilde{p}}(f)$ is the empirical expectation and is defined as:

$$
E_{\tilde{p}}(f) = \sum_{h,s} \tilde{p}(h, s) f(h, s)
\tag{5}
$$

Under these constraints, ME principle guarantees a learned model as uniform as possible, and the model can be obtained by maximizing the conditional entropy of the training data:

$$H(p) = -\sum_{h,s} \tilde{p}(h)p(s \mid h)\log p(s \mid h) \tag{6}$$

It results in the probability function of exponential form:

$$p(s \mid s', o) = \frac{1}{Z(h, s')}\exp(\sum_i \lambda_i f_i(h, s)) \tag{7}$$

where λ is the weight of the feature f_i, and Z is the normalization factor.

In the above formula, $p(s \mid s', o)$ associates observation with state transition, which makes data sparseness a serious problem. Therefore, a variant form of MEMM is proposed. It makes observations associated with state instead of state transition. Then, MEMM is decomposed into two sub-models: the ngram model and the ME model. The probability function is reformulated as:

$$p(s \mid s', o) = p(s \mid s')p_{ME}(s \mid h) \tag{8}$$

where $p_{ME}(s \mid h)$ is the conditional probability which is estimated under the ME principle and has the exponential form. Since the data sparseness problem is prone to occur in the Pinyin-to-Character Conversion task, our work is based on Formula (8).

Accordingly, the training process of MEMM can be decomposed into two separate processes for the ngram model and the ME model. The ngram model can be effectively trained by the Maximum Likelihood Estimation (MLE) principle [Myung 2003]. For the ME model, there is no easy solution to get the optimal value of λ directly. Some iterative algorithms, *i.e.* the Generalized Iterative Scaling (GIS) algorithm [Darroch and Ratcliff 1972] and the Improved Iterative Scaling (IIS) algorithm [Pietra *et al*. 1997], are usually adopted. However, the time complexity of the iterative algorithm is far beyond the complexity of the MLE method, and it becomes the bottleneck of the training process of MEMM. When the scale is large, it is infeasible to use the iterative algorithm to train the MEMM model because of the high complexity.

3. Principle of Class-Based MEMM

This paper involves the class of state in traditional MEMM so as to address its efficiency problem on a large scale of state set. A Class-based MEMM model is proposed and its probability functions are strictly deduced and well formulized both in the case of hard class and soft class. The section is structured as follows. First, it presents C-MEMM in the case of

hard class. Second, it describes C-MEMM in the case of soft class. Third, it provides ways to get the class of the state.

3.1 C-MEMM on Hard Class

The simplest way to construct C-MEMM is to substitute state with class of state in the probability of $p_{ME}(s \mid h)$ in Formula (8). Then, $p_{ME}(c \mid h)$ is used to simulate $p_{ME}(s \mid h)$ in which c is the class of s. However, the predicative capability of $p_{ME}(c \mid h)$ is much lower than that of $p_{ME}(s \mid h)$, which decreases the overall performance of C-MEMM. This paper begins the work from calculating the conditional probability of sequential data, and re-deduces the probability functions for C-MEMM according to the Bayes rule and the Markov property. More formally, the following notations are defined:

- $O = o_1, o_2 ... o_n$: the observation sequence
- $S = s_1, s_2 ... s_n$: the state sequence
- $C = c_1 c_2 ... c_n$: the class sequence which corresponds to S. It is unique in the case of hard class.

In the case of hard class, where the class sequence is completely determined by the state sequence, the following equation can be made:

$$P(S \mid O) = P(S, C \mid O) . \tag{9}$$

Then, the probability function of C-MEMM can be deduced through the following process:

$$P(S \mid O) = P(S, C \mid O) \overset{Bayesian\ Rule}{=} P(C \mid O) \times P(S \mid C, O) . \tag{10}$$

According to the Bayes rule, the conditional probability of sequential data is decomposed into two conditional probabilities. The probability of $P(C \mid O)$ can be further decomposed by the Bayes rule and the Markov property, exactly as the process of Formula (1). The ultimate formula is directly presented as below:

$$P(C \mid O) = p(c_1 \mid o_1) \prod_{i=2}^{n} p(c_i \mid c_{i-1}, o_i) = p(c_1 \mid o_1) \prod_{i=2}^{n} p(c_i \mid c_{i-1}) p_{ME}(c_i \mid h_i) \tag{11}$$

In the above formula, $p(c_i \mid c_{i-1}, o_i)$ is further decomposed by Formula (8).

For the probability of $P(S \mid C, O)$, the decomposing process is a little more complex and an additional independent assumption should be made.

$$
\begin{aligned}
P(S \mid C,O) &= p(s_1 s_2 \ldots s_n \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n) \\
&\quad \textit{Bayesian Rule} \\
&= p(s_1 \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n) \times p(s_2 \ldots s_n \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n, s_1) \\
&\quad \textit{Markov Property} \\
&= p(s_1 \mid c_1, o_1) \times p(s_2 \ldots s_n \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n, s_1) \\
&\quad \textit{Bayesian Rule} \\
&= p(s_1 \mid c_1, o_1) \times p(s_2 \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n, s_1) \times p(s_3 \ldots s_n \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n, s_1 s_2) \\
&\quad \textit{Markov Property} \\
&= p(s_1 \mid c_1, o_1) \times p(s_2 \mid c_2, o_2, s_1) \times p(s_3 \ldots s_n \mid c_1 c_2 \ldots c_n, o_1 o_2 \ldots o_n, s_1 s_2) \\
&\quad \ldots\ldots \\
&= p(s_1 \mid c_1, o_1) \times \prod_{i=2}^{n} p(s_i \mid c_i, o_i, s_{i-1}) \\
&\quad \textit{Independent Rule} \\
&= p(s_1 \mid c_1, o_1) \times \prod_{i=2}^{n} p(s_i \mid c_i, o_i) \times p(s_i \mid s_{i-1})
\end{aligned}
\tag{12}
$$

The fore part of the above deduction is exactly the same as the process in Formula (1). In the following part, such an assumption is made that the state transition probability is independent of the emission probability. The local conditional probability of $p(s_i \mid c_i, o_i, s_{i-1})$ is then decomposed into two probabilities: $p(s_i \mid s_{i-1})$ and $p(s_i \mid c_i, o_i)$, in which $p(s_i \mid s_{i-1})$ is the state transition probability and $p(s_i \mid c_i, o_i)$ is the class-based emission probability. To gain more insight, $p(s_i \mid c_i, o_i)$ can be rewritten as $p(s_i \mid o_i, c_i)$ which is the emission probability that is conditioned on the class of c_i. Together with the decompositions of Formulas (10) and (11), the ultimate form of the probability function of C-MEMM can be obtained, presented as below:

$$
P(S \mid O) = p(c_1 \mid o_1) p(s_1 \mid c_1, o_1) \prod_{i=2}^{n} p(c_i \mid c_{i-1}) p_{ME}(c_i \mid h_i) p(s_i \mid c_i, o_i) p(s_i \mid s_{i-1}) \tag{13}
$$

Until now, the conditional probability of sequential data has been decomposed into four kinds of local conditional probabilities. The probability of $p_{ME}(c_i \mid h_i)$ is estimated under the ME principle and it has the exponential form:

$$
p_{ME}(c \mid h) = \frac{1}{Z(h)} \exp(\sum_i \lambda_i f_i(h, c)) \,. \tag{14}
$$

As the scale of c is much smaller than that of s, it needs shorter time for C-MEMM to estimate $p_{ME}(c_i \mid h_i)$ than $p_{ME}(s_i \mid h_i)$, which makes it feasible to apply C-MEMM in the tasks with large scale of state set, *i.e.* the Pinyin-to-Character Conversion task. The other three probabilities are estimated by the Maximum Likelihood Estimation (MLE) principle, presented as below:

$$p(c_i \mid c_{i-1}) = \frac{C(c_{i-1}, c_i)}{C(c_{i-1})} \tag{15}$$

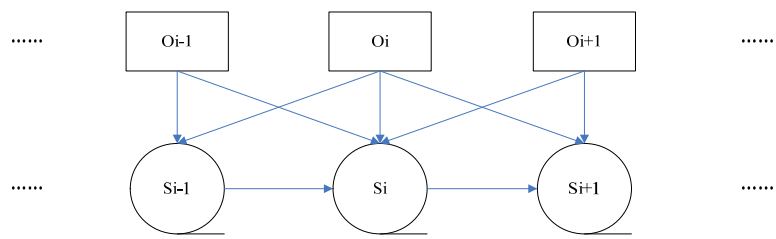
where $C(x)$ is the occurrence times of x in the training corpus.

$$p(s_i \mid s_{i-1}) = \frac{C(s_{i-1}, s_i)}{C(s_{i-1})} \tag{16}$$

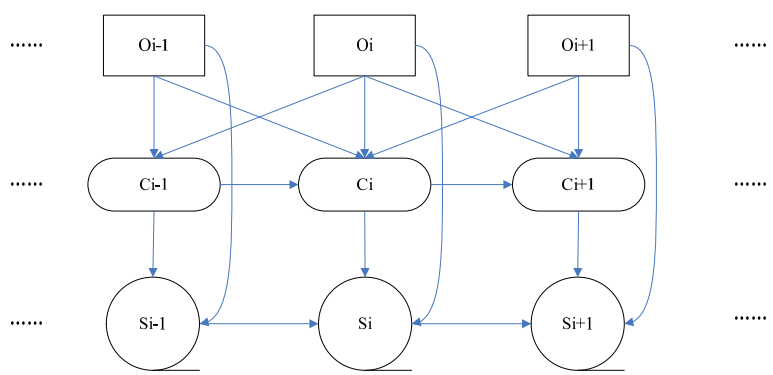
$$p(s_i \mid c_i, o_i) = \frac{C(c_i, o_i, s_i)}{C(c_i, o_i)} \tag{17}$$

When applying C-MEMM in the Pinyin-to-Character Conversion tasks, the four kinds of local conditional probabilities are first estimated from the training corpus. Then, according to the input pinyin sequence, the probability of a character sequence candidate is calculated by Formula (13). Finally, the most probable character sequence is selected as the conversion results for the input pinyin sequence. Some dynamic programming algorithms can be utilized in the above process, *i.e.* the Viterbi algorithm.

In the remaining part of this section, the probability dependency graph in C-MEMM is presented and an intuitional description is provided on the functions of the four local conditional probabilities.



(a) Dependency Graph for MEMM



(b) Dependency Graph for C-MEMM

Figure 1. Probability Dependency Graphs for MEMM and C-MEMM

Presented as the above graphs, the constraints from the observation sequence are imposed directly on the state sequence in MEMM. The scale of the state set becomes a bottleneck in the training process of MEMM. However, in C-MEMM, there is a class sequence between the state sequence and the observation sequence. All the constraints from the observation sequence are imposed on the class sequence in C-MEMM, rather than directly on the state sequence as in MEMM. Since the scale of the class set is much smaller than that of the state set, the conditional probability of $p_{ME}(c_i \mid h_i)$, which connects the observation sequence with the class sequence, can be efficiently estimated under the ME principle. The constraints from the observation sequence are also well exploited by the probabilities of the class sequence. Furthermore, all the constraints from the observation sequence are conveyed from the class sequence into the state sequence by the conditional probabilities between these two sequences.

Concretely speaking, in Formula (13), the conditional probability of $p_{ME}(c_i \mid h_i)$ and the class transition probability of $p(c_i \mid c_{i-1})$ aim to model the constraints from the observation sequence and conserve them in the probability of the class sequence. The conditional probability of $p(s_i \mid c_i, o_i)$ conveys these constraints from the class sequence to the state sequence. The three conditional probabilities, together with the state transition probability of $p(s_i \mid s_{i-1})$, form the probability function of C-MEMM.

Moreover, since there is rich syntactic and semantic information in word class [Brown *et al*. 1992], C-MEMM used in the Pinyin-to-Character Conversion task can well utilize this additional information to realize further improvement.

3.2 C-MEMM on Soft Class

In the above section, the probability function of C-MEMM is deduced in the case of a hard class in which the state is restricted to only one class. However, in natural language processing tasks, *i.e*. the Pinyin-to-Character Conversion task, the state of C-MEMM is usually defined as word in the lexicon which usually belongs to multiple classes in nature. For example, part-of-speech (POS) can be taken as a natural hierarchy of word class. Most words possess more than one kind of POS tag. Each POS tag represents a certain syntactic and semantic property of the word. It is beneficial for C-MEMM to exploit all the properties of the word in natural language processing. The section studies C-MEMM in the case of soft class in which the state belongs to multiple classes. The probability function is re-deduced for C-MEMM.

In the case of a soft class, there are many class sequences corresponding to one state sequence. In order to calculate the probability of the state sequence, the conditional probabilities of all the class sequences should be summarized. Therefore, it is more complex to deduce the probability function of C-MEMM in the case of soft class than hard class. Similar to the case of the hard class, this section begins the work from calculating the

conditional probability of sequential data, presented as below:

$$P(S \mid O) = \sum_{C} P(S, C \mid O) \overset{Bayesian\ Rule}{=} \sum_{C} P(C \mid O) \times P(S \mid C, O) . \quad (18)$$

The decompositions of $P(C \mid O)$ and $P(S \mid C, O)$ are exactly the same as those in the case of the hard class, which were presented in the above section. Then, the probability function in the case of soft class can be directly described as below:

$$P(S \mid O) = \sum_{c_1 \ldots c_n} \{p(s_1 \mid c_1, o_1) p(c_1 \mid o_1) \prod_{i=2}^{n} p(c_i \mid c_{i-1}) p_{ME}(c_i \mid h_i) p(s_i \mid c_i, o_i) p(s_i \mid s_{i-1})\} . \quad (19)$$

$p_{ME}(c_i \mid h_i)$, $p(c_i \mid c_{i-1})$, $p(s_i \mid c_i, o_i)$ and $p(s_i \mid s_{i-1})$ are estimated exactly in the same way as in the hard class. The probability dependency graph in the case of soft class is presented as below:

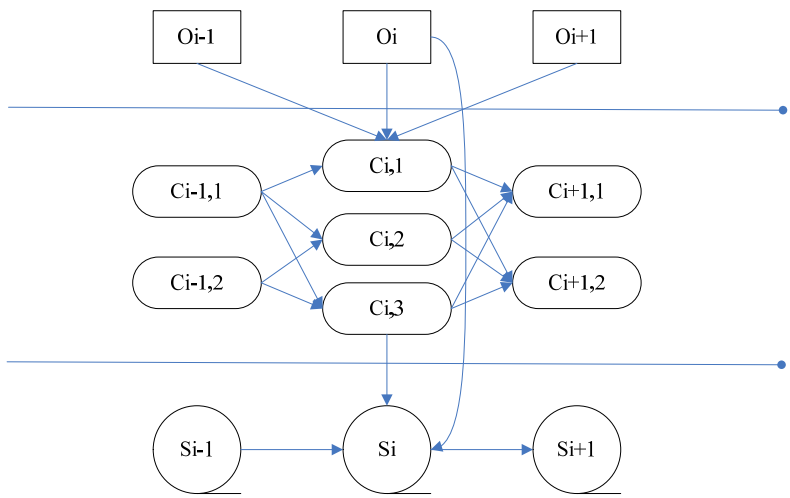


Figure 2. Probability Dependency Graph for C-MEMM on Soft Class

Differing from the case of a hard class, there are multiple class sequences between the observation sequence and the state sequence in the case of a soft class. In order to calculate the probability of $P(S \mid O)$, it is necessary to summarize all the conditional probabilities in these class sequences. The time complexity increases at an exponential rate with the length of sequence. Some dynamic algorithms, *i.e.* the forward algorithm and the backward algorithm, can calculate $P(S \mid O)$ efficiently at the polynomial time complexity. However, in the Pinyin-to-Character Conversion task, it is to find the optimal state sequence of S which maximizes the probability of $P(S \mid O)$. This is the *decoding problem of C-MEMM*. In a straightforward way, it's necessary to enumerate all the possible sequences of S and calculate the value of $P(S \mid O)$ for each sequence. The optimal sequence with the highest $P(S \mid O)$ is

then selected from them. In reality, it is infeasible because of the high time complexity. The dynamic algorithm, *i.e*. the Viterbi algorithm, is expected to solve the decoding problem. However, Formula (19) makes a global summarization in the class sequences in which the Viterbi algorithm can not be applied. A simplification is then made in this paper. The global summarization, which is based on the whole sequence of class, is decomposed into the local summarization which is only based on the class at certain position. The probability function is simplified as below:

$$P(S \mid O) \approx \sum_{c_1} p(s_1 \mid c_1, o_1) p(c_1 \mid o_1) \times \prod_{i=2}^{n} \sum_{c_i} p(c_i \mid c_{i-1}) p_{ME}(c_i \mid h_i) p(s_i \mid c_i, o_i) p(s_i \mid s_{i-1}) \quad .(20)$$

The Viterbi algorithm can be applied to Formula (20) and can find the optimal state sequence of S in a polynomial time complexity. The dependency relationship graph is then described as below:

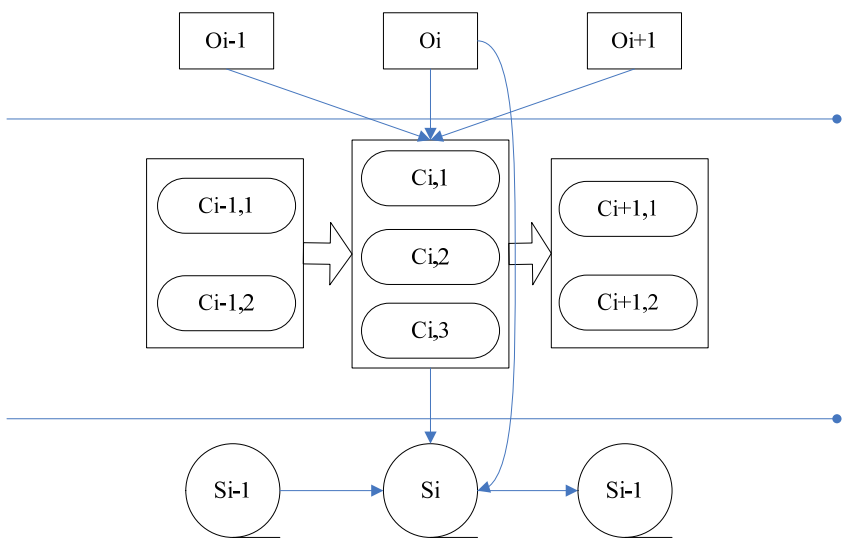


Figure 3. Probability Dependency Graph for the Simplified C-MEMM on Soft Class

3.3 Hierarchy of State Class

There are two ways to get the class of state. One is the statistical method, by which the state class is obtained by the clustering algorithm from the training corpus. However, according to Zip's law, there are always low-frequency or zero-frequency states in the training corpus. Their frequencies are not statistically significant, and they can not be properly clustered by the statistical methods. The other method is getting the class from the pre-defined thesaurus. The hierarchy of class is defined by linguists according to the syntax and semantic information of each state. It can be taken as the well-defined hierarchy of state class. This paper attains the

hierarchy of state class in the second way. TongyiciCilin is adopted as the hierarchy of state class in the case of hard class and the set of POS tag is adopted in the case of soft class. The detailed information is presented in Section 4.1.

4. EXPERIMENTS AND DISCUSSIONS

This section evaluates C-MEMM in the Pinyin-to-Character Conversion task. First, the data set is described. Second, the experimental results are presented. The performances of C-MEMM are evaluated both in the case of hard class and soft class. Third, the conclusion is drawn.

4.1 Data Set Description

This section describes the data set used in the experiments. First, information about the text corpus is presented. Then, the way to get pinyin corpus is described. Finally, the hierarchies of word class are presented.

Text Corpus

This paper chooses six months of the People's Daily corpus in 1998 as the text corpus in the experiments. The corpus has been annotated by Peking University with the POS tags and the name entities [Yu *et al*. 2003]. It has become the standard corpus in Chinese language processing in recent years [Emerson 2005]. There are 46 kinds of POS tag in the POS set. They are listed in Table 1:

Table 1. POS Set of Peking University

POS Set of Peking University							
Ag	*a*	*ad*	*an*	*Bg*	*b*	*c*	*Dg*
d	*e*	*f*	*g*	*h*	*i*	*j*	*k*
l	*Mg*	*m*	*Ng*	*n*	*nr*	*ns*	*nt*
nx	*nz*	*o*	*p*	*Qg*	*q*	*Rg*	*r*
s	*Tg*	*t*	*Ug*	*u*	*Vg*	*v*	*vd*
vn	*w*	*x*	*Yg*	*y*	*z*		

The text corpus is divided into two parts: the training corpus which consists of the first five months' corpora, and the testing corpus which is the sixth month's corpus. The detailed information is presented in Table 2:

Table 2. Description of the Text Corpus

	Training Corpus	Testing Corpus
Number of months	*1-5 months*	*6^{th} month*
Number of characters	*9.09×10^{6}*	*1.88×10^{6}*

Pinyin Corpus

The pinyin corpus is necessary for evaluating C-MEMM in the Pinyin-to-Character Conversion task. When C-MEMM is evaluated, the pinyin corpus is first converted into the character corpus by C-MEMM. Then, the conversion results are compared with the standard text corpus and the error rate is calculated. The pinyin corpus is obtained from the text corpus by a conversion toolkit[1] which achieves 99.7% accuracy on a golden pinyin corpus. In the experiments, the errors in the pinyin corpus could lead to the conversion error of C-MEMM. Therefore, the actual error rate of C-MEMM is a little lower than the reported results in this paper. However, there are not many errors in the pinyin corpus because of the high precision of our conversion toolkit. Thereby, the experimental results can be regarded to be close enough to the actual performance of C-MEMM.

Hierarchy of Word Class

Moreover, word class is necessary for building up C-MEMM in the Pinyin-to-Character Conversion task. The paper gets the hierarchy of word class from the compiled thesaurus which contains the word class information.

TongyiciCilin [Mei *et al*. 1983] is adopted as the hierarchy of word class in the experiments of hard class. TongyiciCilin was initially complied in 1982. There were initially 5.38×10^4 words which were organized into a tree structure according to their syntax and semantic information. The structure is shown in Figure 4. There are a total of four layers in the tree. The word is represented by the leaf node in the leaf layer. The word class is represented by the internal node in the internal layer. There is a road from each leaf node to the root node. On the road, there are several internal nodes of different layers which represent the classes of different scales that the leaf node belongs to. The node in the higher layer represents the class of bigger scale which usually corresponds to a more general concept of Chinese, and vice-versa. Each layer represents a pattern of word class, and the nodes in the same layer describe a way to cluster the words in TongyiciCilin. Moreover, the lower the layer is, the finer the word classes are, therefore the more syntactic and semantic information the layer contains. For example, the 3rd layer contains more syntactic and semantic information than the 1st layer does in Figure 4.

[1] http://www.insun.hit.edu.cn/product/viewproduct.asp?id=105

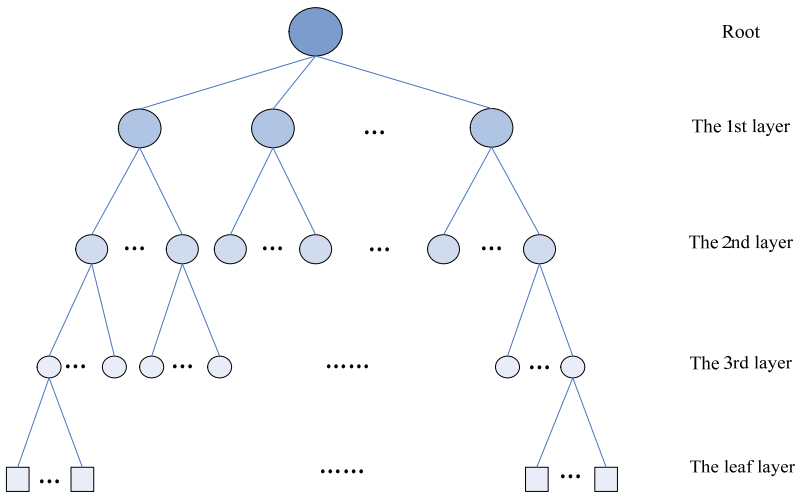


Figure 4. Hierarchy of Word Class in TongyiciCilin

In recent years, an extended version [Liu *et al*. 2005] of the original TongyiciCilin has been complied. Some infrequent words have been deleted, while some new words have been added. The scale of the lexicon in the new version is up to 7.73×10^4. The detailed information is described in Table 3:

Table 3. Description of TongyiciCilin (new version)

Description of TongyiciCilin	
Scale of lexicon	7.73×10^4
Number of Cluster in 1st layer	*12*
Number of Cluster in 2nd layer	*97*
Number of Cluster in 3rd layer	*1428*

This paper adopts the new version of TongyiciCilin in the experiments of hard class.

In the experiments of soft cluster, the POS set is a natural choice for the hierarchy of word class. The information of the POS set has been provided in the beginning of this section.

4.2 Experiments on the Hard Class

This section investigates C-MEMM in the case of hard class in the Pinyin-to-Character Conversion task. TongyiciCilin is adopted as the hierarchy of word class. All the words in TongyiciCilin are adopted as the lexicon. The bigram model is taken as the baseline model. The additive smoothing technique is utilized. One order of C-MEMM is evaluated. Ten feature types of the pinyin constraints are extracted and exploited in C-MEMM. They are listed in Table 4:

Table 4. Feature Types in C-MEMM

	Feature Type	Feature Description
Atomic Feature Type	Yin_i	*The current pinyin*
	Yin_{i-1}	*The previous pinyin*
	Yin_{i-2}	*The previous but one pinyin*
	Yin_{i+1}	*The next pinyin*
	Yin_{i+2}	*The next but one pinyin*
	$YinComb_i$	*The pinyin combination of the current word which usually consists of several pinyin strings.*
Combined Feature Type	$Yin_i\ Yin_{i-1}$	*The combination of Yin_i and Yin_{i-1}*
	$Yin_i\ Yin_{i+1}$	*The combination of Yin_i and Yin_{i+1}*
	$Yin_{i-1}\ Yin_{i-2}$	*The combination of Yin_{i-1} and Yin_{i-2}*
	$Yin_{i+1}\ Yin_{i+2}$	*The combination of Yin_{i+1} and Yin_{i+2}*

From the above feature types, two feature templates are constructed so as to investigate the effectiveness of different feature types in C-MEMM performances. In template one, the size of the context window is set to 3, based on which the model of C-MEMM-1 is constructed. In template two, the size of the context window is set to 5, based on which the model of C-MEMM-2 is constructed. The information is presented in Table 5:

Table 5. Feature Templates in C-MEMM

Feature Template	Feature Types	Model
Template One	Yin_i, Yin_{i-1}, Yin_{i+1}, $YinComb_i$, $Yin_i\ Yin_{i-1}$, $Yin_i\ Yin_{i+1}$	*C-MEMM-1*
Template Two	Yin_i, Yin_{i-1}, Yin_{i-2}, Yin_{i+1}, Yin_{i+2}, $YinComb_i$, $Yin_i\ Yin_{i-1}$, $Yin_i\ Yin_{i+1}$, $Yin_{i-1}\ Yin_{i-2}$, $Yin_{i+1}\ Yin_{i+2}$	*C-MEMM-2*

As mentioned above, there are several ways to cluster a word in TongyiciCilin, each corresponding to an internal layer in the tree structure of TongyiciCilin. C-MEMM is built up based on each pattern of word class used separately for each internal layer in TongyiciCilin. The performance of C-MEMM is investigated and the error rates are presented in Table 6:

Table 6. Error Rate of C-MEMM in the case of Hard Class

	No cluster	Clusters of 1st layer	Clusters of 2nd layer	Clusters of 3rd layer
Baseline	9.15%	---	---	---
C-MEMM-1	---	6.10%	**5.84%**	5.85%
Reduction	---	33.33%	**36.17%**	36.07%
C-MEMM-2	---	5.73%	5.46%	**5.28%**
Reduction	---	37.38%	40.33%	**42.30%**

The error rate of the baseline model is presented in the category of 'No cluster' from which the error rate reductions are calculated. According to the experimental results, C-MEMM outperforms the baseline model significantly with great error rate reduction. As much as 36.17% reduction has been achieved by C-MEMM-1 and 42.30% reduction has been yielded by C-MEMM-2. It proves that the predicative capability of C-MEMM is superior to that of the ngram model in the Pinyin-to-Character Conversion task. In addition, comparing the performance of C-MEMM-1 with C-MEMM-2, C-MEMM-2 outperforms C-MEMM-1 slightly, due to modeling the richer feature types of the pinyin constraints. This fact proves that the improvements of C-MEMM in the Pinyin-to-Character Conversion task are due to the exploitation of the pinyin constraints from the input pinyin sequence. Finally, the section investigates the performance of C-MEMM based on different patterns of word class. As mentioned in the above section, there is an increase of syntactic and semantic information contained in the word classes from the 1st internal layer to the 3rd internal layer of TongyiciCilin. From Table 6, it can be found that the error rates of C-MEMM generally decrease from the 1st layer to the 3rd layer, which proves that C-MEMM can make good use of the syntactic and semantic information from the word classes and attain further improvement.

To draw a conclusion, C-MEMM achieves significant error rate reductions from the ngram model in the Pinyin-to-Character Conversion task by exploitation of pinyin constraints. In addition, C-MEMM makes good use of the syntactic and semantic information in word class and sees further improvement.

4.3 Experiments on the Soft Class

This section evaluates C-MEMM in the case of soft class. The POS set of Peking University is taken as the hierarchy of word class. A word list compatible with the POS set is adopted as the lexicon. Other settings are the same as those in the case of hard class. The experimental results are presented in Table 7:

Table 7. Error Rate of C-MEMM in the case of Soft Class

	Baseline	C-MEMM-1	C-MEMM-2
Error rate (%)	8.37%	6.00%	5.82%
Reduction (%)	------	28.15%	30.47%

The experimental results are similar to the results in the case of hard class. First, C-MEMM outperforms the baseline model significantly. As much as 28.15% error rate reduction was achieved by C-MEMM-1 and a 30.47% error rate reduction was obtained by C-MEMM-2. This proves that C-MEMM is much more powerful than the ngram model. Second, C-MEMM-2 gets better performance than C-MEMM-1, due to modeling the richer feature types of the pinyin constraints. This indicates that the improvements of C-MEMM are

due to the exploitation of the input pinyin information. Therefore, the conclusion is drawn that C-MEMM (hard-class based or soft-class based) improves the performance of the Pinyin-to-Character Conversion system significantly by exploitation of the pinyin constraints from the pinyin sequence.

In the remaining part of this section, the performance of the soft-class based MEMM is compared with the hard-class based MEMM. However, the experimental results in this section can not be compared directly with the results in Section 4.2, due to the fact that different lexica were used in the two sections. For fair comparison, a hierarchy of hard class is created from the hierarchy of soft class in this section. It restricts only one POS tag for each word in the lexicon. The most frequent POS tag of that word is adopted in the hierarchy of hard class. The experimental results are presented in Table 8:

Table 8. Comparison between Soft-class based MEMM and Hard-class based MEMM

	Baseline	C-MEMM-1	C-MEMM-2
No class	8.37%	------	------
Hard class	------	6.21%	6.17%
Soft class	------	6.00%	5.82%

As shown in Table 8, the soft-class based MEMM performs better than the hard-class based MEMM to some extent, proving that the soft-class based MEMM can exploit the comprehensive properties of word to achieve better performance.

4.4 Comparison with Class-based Ngram Model

The class-based ngram model enhances the traditional ngram model by involving word class [Brown *et al*. 1992]. The data sparseness problem is alleviated, while the syntactic information is captured by word class. The motivation and the formulation of the class-based ngram model are similar to those of C-MEMM. Therefore, this section compares the performances of C-MEMM with those of the class-based ngram model.

First, this section compares the performance of the hard-class based MEMM with that of the class-based ngram model. The traditional bigram model is taken as the baseline model. Several class-based ngram models are built up according to the word class pattern of each internal layer of TongyiciCilin. The experimental results are presented in Table 9:

Table 9. Comparison between Hard-class based MEMM and Class-based Ngram Model

	No cluster	Clusters of 1st layer	Clusters of 2nd layer	Clusters of 3rd layer
Baseline	9.15%	---	---	---
C-Ngram	---	8.25%	7.74%	**7.37%**
C-MEMM-1	---	6.10%	**5.84%**	5.85%
C-MEMM-2	---	5.73%	5.46%	**5.28%**

From Table 9, the class-based ngram models achieve lower error rates than the baseline model, showing a more powerful predicative capability. What's more, the error rates of the class-based ngram models decrease from the 1st layer to the 3rd layer, proving that the improvement of the class-based ngram model is due to the exploitation of the increasing syntactic and semantic information of word class. However, the class-based ngram models underperformed the hard-class based MEMM models. The latter can not only make use of the syntactic and semantic information of word classes but also exploit the pinyin constraints from the input pinyin sequences.

In the following, the performance of the soft-class based MEMM with that of the class-based ngram model is compared. The POS ngram is constructed and interpolated with the traditional word ngram model. The experimental results are presented in Table 10:

Table 10. Comparison between Soft-class based MEMM and Class-based Ngram Model

	Baseline	C-Ngram	C-MEMM-1	C-MEMM-2
Error rate	8.37%	7.89%	6.00%	5.82%

The experimental results are similar to those found in Table 9. The class-based ngram models outperform the traditional ngram model by exploitation of the syntactic and semantic information in word class. However, they underperformed the soft-class based MEMM models because the latter could also make use of the pinyin constraints from pinyin sequence.

In conclusion, both the C-MEMM model and the class-based ngram model can make good use of the syntactic and semantic information of word class so as to improve the performance in the Pinyin-to-Character Conversion task; however, the former outperforms the latter by additionally exploiting the pinyin constraints from the pinyin sequence.

5. Related Works

To the best of our knowledge, there is no literature that proposes a class expansion to the MEMM model. John Lafferty [Lafferty and Suhm 1996] proposes a cluster expansion to the GIS algorithm so as to train the ME language model efficiently. However, as Lafferty admits,

the technique is of little use in computing the exact ME solution. Joshua Goodman [Goodman 2001] proposes a speedup technique for the ME training process in language modeling. He decomposes the traditional ngram model into several class-based ngram models and applies the ME principle in each sub-model. There are significant differences between the Goodman 's work and this paper's. First of all, C-MEMM aims to solve the sequence label problem instead of the sequence ranking problem as language model does. Second, this paper deduces the probability function of C-MEMM based on the conditional probability of the *whole* sequence, whereas Goodman gets the probability function based on the decomposition of the *local* ngram probability. Third, this paper applies C-MEMM to the Pinyin-to-Character Conversion task in order to improve the application performance; however, Goodman is used to speed up the training process of the ME model. Moreover, both the case of hard class and soft class are discussed in this paper. In contrast, Goodman's technique is built up only in the case of a hard class.

6. Conclusions

This paper aims to improve the performance of the Pinyin-to-Character Conversion system by exploitation of the pinyin constraints from the pinyin sequence. The MEMM framework is used to describe both the pinyin constraint and the character constraint. The Class-based Maximum Entropy Markov Model (C-MEMM) is proposed to solve the efficiency problem of MEMM in the Pinyin-to-Character Conversion task. The probability functions of C-MEMM are strictly deduced and well formulized by the Bayes rule and the Markov property. Both the case of hard class and soft class are well discussed. From the experimental results, the conclusions can be drawn as follows:

- Compared with the traditional ngram model, C-MEMM improves the performance of the Pinyin-to-Character Conversion system effectively by exploitation of the pinyin constraints from the input pinyin sequences.
- C-MEMM can make good use of the syntactic and semantic information in word class and attain further improvement.
- The soft-class based MEMM outperforms the hard-class based MEMM by exploitation of more comprehensive properties of word.

Acknowledgements

This investigation is supported by the key project of the National Natural Science Foundation of China (grant No.60435020), the project of the National Natural Science Foundation of China (grant No.60673037), the project of the High Technology Research and Development Program of China (grant No.2006AA01Z197), the project of MOE-MS Key Laboratory of

Natural Language Processing and Speech in China (grant No.01307620).

Especially, the authors thank the anonymous reviewers for their valuable suggestions and comments.

References

Berger, A, S. D. Pietra, and V. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics,* 1996, 22(1), pp. 39-71.

Brown, P. F., V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language", *Computational Linguistics*, 1992, 18(4), pp. 467-479.

Chen, Y., "Chinese Language Processing", *Shang Hai education publishing company*. 1997

Chen, L. Z. and T. Y. Huang, "A Novel Word Clustering Algorithm And Vari-gram Language Model", *Journal of Computer Sciences*, 1999, 22(9), pp. 942-948.

Chen, Z, and K. F. Lee, "A New Statistical Approach To Chinese Pinyin Input", *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL2000)*, Hong Kong, 3-6 October 2000.

Darroch, J. N. and D. Ratcliff, "Generalized Iterative Scaling for Log-linear Models". *Annals of Mathematical Statistics*, 1972, 43, pp. 1470-1480.

Emerson, T. "The Second International Chinese Word Segmentation Bakeoff". *In Proceedings of The Fourth Sighan Workshop on Chinese Language Processing*, 2005, pp. 123-133.

Gao, J. F., J. Goodman, and J. B. Miao, "The Use of Clustering Techniques for Language Modeling - Application to Asian languages". *International Journal of Computational Linguistics and Chinese Language Processing,* 6(1), pp. 27-60. 2001.

Gao, J. F, J. Goodman, M. J. Li, K. F. Lee, "Toward a unified approach to statistical language modeling for Chinese", *ACM Transactions on Asian Language Information Processing*, 2002, 1(1), pp. 3-33.

Gao, J. F, H. Yu and W. Yuan, "Minimum Sample Risk Methods for Language Modeling". *In Human Language Technology Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Oct 6-8, Vancouver, Canada, 2005.

Goodman, J., "Classes for fast maximum entropy training". *Proceedings of the IEEE International Conference on Acoustics, Speach and Signal Processing (ICASSP-2001)*, IEEE press, 2001.

Hsu, W. L. and K. J. Chen, "The Semantic Analysis in GOING - An Intelligent Chinese Input System", *Proceedings of the Second Joint Conference of Computational Linguistics*, Shia men, 1993, pp. 338-343.

Hsu, W. L., "Chinese parsing in a phoneme-to-character conversion system based on semantic pattern matching", *International Journal on Computer Processing of Chinese and Oriental Languages,* 1995, volume 40, pp. 227-236.

ISO, "Information and documentation - Romanization of Chinese", ISO 7098:1991

Jeffreys, H, "Theory of Probability". *Clarendon Press*, Oxford, second Edition. 1948.

Jelinek, F. and R. L. Mercer, "Interpolated estimation of markov source parameters from sparse data". *In Pattern Recognition in Practice,* 1980, pp. 381-397.

Katz, S. M. "Estimation of probabilities from sparse data for the language model component of a speech recognizer", *IEEE Transactions on Acoustics, Speeech and Signal Processing,* 1987, 35(3), pp. 400-401.

Kuo, J. J., "Phonetic-input-to-character conversion system for Chinese using syntactic connection table and semantic distance", *Computer Processing and Oriental Languages,* 1995, 10(2), pp. 195-210.

Lafferty, J. and B. Suhm. "Cluster expansions and iterative scaling for maximum entropy language models". *Maximum Entropy and Bayesian Methods*, K. Hanson and R. Silver, eds., Kluwer Academic Publishers, 1996.

Li, H., "Word Clustering and Disambiguation Based on Co-occurrence Data", *Proceedings of COLING-ACL98*. Montreal,Canada. 10-14 August, 1998.

Liu, T. et al., "TongYiCiCiLin (Extension Version)". 2005. Http://www.ir-lab.org

McCallum, A., D. Freitag and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation", *Proceedings of ICML2000*, Stanford, CA, USA, 2000, pp. 591-598.

Mei, J. J, Y. M. Zhu, Y. Q. Kao, H. X. Yan. "TongYiCiCiLin". Shanghai: *Shanghai Lexicographical Publishing House*. 1983.

Myung I. J, "Tutorial on Maximum Likelihood Estimation", *Journal of Mathematical Psychology,* 2003, 47, pp. 90-100.

Pietra, S. D, V. D. Pietra and J. Lafferty. "Inducing Features of Random Fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1997. 19(4), pp. 380-393.

Tsai, J. L. and W. L. Hsu, "Applying an NVEF Word-Pair Identifier to the Chinese Syllable-to-Word Conversion Problem", *Proceedings of COLING02*, Taipei, 2002.

Tsai, J. L., T. J. Chiang and W. L. Hsu, "Applying Meaningful Word-Pair Identifier to the Chinese Syllable-to-Word Conversion Problem", *Proceedings of ROCLING04,* 2004.

Tsai, J. L., "Applying a Mix Word-Pair Identifier to the Chinese Syllable-to-Word Conversion Problem", *In 2th International Joint Conference on Natural Language Processing (IJCNLP 2005)*, Jeju, Korea, Oct 11-13, 2005.

Tsai, J. L., "Using Word Support Model to Improve Chinese Input System", *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING -ACL06)*, Sydney, Australia, 17-21 July 2006.

Wang, X. L., "Chinese Input system by Pinyin Sentence: Insun", *Journal of Chinese Information Processing*, 1993, 7(2), pp. 45-54.

Wang, X. L, Q. C Chen and D. S. YEUNG, "Mining pinyin-to-character conversion rules from large-scale corpus:a rough set approach", *IEEE Transaction on Systems Man and Cybernetics*, Part B: Cybernetics, 2004, 34(2), pp. 834-844.

Wang, X. L., D. S . Yeung, J. N. K. Liu, R. W. P. Luk and X. Wang, "A Hybrid Language Model Based on Statistics and Linguistic Rules", *International Journal of Pattern Recognition and Artificial Intelligence*, 2005, 19(1), pp. 109-128.

Wang, Y. M., "The Three Principles of Computer Chinese Character Keyboard Design", *Chinese Journal of Computers*, 2005, 28(5), pp. 870-881.

Wu, J., "Implementation and Application of Statistical Language Model in Mandarin Speech Recognition", *master dissertation of the Tsinghua University,* 2000.

Xiao, J. H, B. Q Liu and X. L. Wang, 2005a, "Principles of Non-stationary Hidden Markov Model and its Applications on Sequence Labeling Task". *In Proceedings of the 2th International Joint Conference on Natural Language Processing (IJCNLP 2005)*, Jeju, Korea, Oct 11-13, 2005a.

Xiao, J. H., B. Q. Liu and X. L. Wang, 2005b, "A Similarity-based Approach to Data Sparse Problem of Chinese Language Modeling", *In Proceedings of the 4th Mexico International Conference on Artificial Intelligent (MICAI2005)*. pp. 761-769, Best Student Paper Award. Mexico, November 14-18, 2005b.

Xiao, J. H., B. Q. Liu and X. L. Wang, "A Similarity-Based Smoothing Algorithm for Chinese Language Modeling and its Application on Pinyin-to-Character Conversion", *High Technique Letters*, 2006, 16(2), pp. 127-132.

Xu, Z. M., X. L. Wang and S. X. Jiang, "A Sentence-Level Chinese Character Input Method", *High Technique Letters*, 2000, (1), pp. 51-56.

Yu, S. W., H. M. Duan, B. Swen and B. B. Chang, "Specification for Corpus Processing at Peking University: Word Segmentation, POS Tagging and Phonetic Notation". *Journal of Chinese Language and Computing*, 2003, 13(2), pp. 121-158.

Zhang, R. Q, Z. Y. Wang and J. P Zhang, "Chinese Pinyin-to-Text Translation technique with Error Correction used for Continuous Speech Recognition", *Journal of Tsinghua University*, 1997, 37(10), pp. 9-12.

Zhang, R. Q., Z. Y. Wang and D. J. Lu, "Zero-Probablities of Language Model in Translations of Chinese Spellings to Characters", *Acta Electornica Sinica,* 1998, 26(8), pp. 43-46.

Zhou G. D. and K. T. Lua. "Word Association and MI-Trigger-based Language Modeling". *Proceedings of COLING-ACL98*. Montreal,Canada. 10-14 August, 1998.

Zipf, G. K., "Psycho-Biology of Languages", *The MIT Press*. 1935.