A Machine Translation Approach for Chinese Whole-Sentence Pinyin-to-Character Conversion*

Shaohua Yang and **Hai Zhao**† and **Bao-liang Lu**
MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems,
Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai 200240, China
shyang.ok@gmail.com, zhaohai@cs.sjtu.edu.cn, blu@cs.sjtu.edu.cn

Abstract

This paper introduces a new approach to solve the Chinese Pinyin-to-character (PTC) conversion problem. The conversion from Chinese Pinyin to Chinese character can be regarded as a transformation between two different languages (from the Latin writing system of Chinese Pinyin to the character form of Chinese,Hanzi), which can be naturally solved by machine translation framework. PTC problem is usually regarded as a sequence labeling problem, however, it is more difficult than any other general sequence labeling problems, since it requires a large label set of all Chinese characters for the labeling task. The essential difficulty of the task lies in the high degree of ambiguities of Chinese characters corresponding to Pinyins. Our approach is novel in that it effectively combines the features of continuous source sequence and target sequence. The experimental results show that the proposed approach is much faster, besides, we got a better result and outperformed the existing sequence labeling approaches.

1 Introduction

There are more than twenty thousand different Chinese characters adopted by Chinese language so that it is a difficult task to type the Chinese character directly from a Latin-style keyboard. Chinese Pinyin is such an encoding scheme that can map the Chinese character to a group of Latin letters so that each character usually has an unique Pinyin representation[1]. Pinyin is originally designed as the phonetic symbol of a Chinese character. For example, Pinyin for the Chinese character **"我"**(I,me) is **"wo"**. As one of the most important topic in Chinese natural language process, Pinyin-to-character(PTC) problem refers to the automatic transformation from Chinese Pinyin sequence to Chinese character sequence. It plays an important or even key role in areas such as speech recognition, Chinese keyboard input method and etc.

There are five different tones for Chinese pronunciation. In Chinese Pinyin system, tone is represented as an accent symbol over Latin letters, which is not convenient to input and thus usually ignored in most Chinese keyboard input methods.

The Chinese PTC problem can be very challenging for the following reasons: there are about 410 Pinyins(without considering five different tones), however, there are ten thousands Chinese characters, even the most popular accounts for about 5,000. So it is quite common to see the phenomenon that different Chinese characters have the same Pinyin. On the average, there are about ten or more Chinese characters which are corresponding to one Pinyin.

When longer Pinyin sequence is given, number of the corresponding legal character sequences will be heavily reduced. Thus, to alleviate the ambiguity

*This work was partially supported by the National Natural Science Foundation of China (Grant No. 60903119 and Grant No. 61170114), the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20110073120022, the National Basic Research Program of China (Grant No. 2009CB320901) and the European Union Seventh Framework Programme (Grant No. 247619).

†Corresponding author

[1]A few Chinese characters are pronounced in several different ways, so they may have multiple Pinyin representation.

26th Pacific Asia Conference on Language,Information and Computation pages 333–342

zi	ran	yu	yan	chu	li
字	**然**	与	严	出	**理**
子	染	**语**	眼	除	离
自	燃	于	烟	**处**	力
紫	冉	鱼	**言**	初	李
资	髯	雨	演	触	利

Table 1: One Pinyin can be mapped to multiple Chinese character (the bolded characters are the correct choices corresponding to the Pinyin sequence.

and speedup the process, in a typical Chinese (Latin) keyboard input method, one always try to type as long Pinyin sequence as possible.

In this paper, we consider such a typical PTC task when a whole sentence of Pinyin sequence is given, and we attempt to recover its original character sequence. In detail, the object of the PTC is to find correct character sequence $C = c_1, c_2, ..., c_n$ given a Pinyin sequence $S = s_1, s_2, ..., s_n$ of which s_i refers to the Pinyin character and c_i refers to the Chinese character. For example, Table 1 illustrates the Pinyin sequence "zi ran yu yan chu li" (自然语言处理, natural language processing) and its corresponding Chinese character sequence. From this table we can observe that one Pinyin can be aligned to too many Chinese characters, though only the underlined bolded Chinese characters are the sequence that we actually intent to get. For example, the Pinyin "zi" can be mapped to Chinese characters include "字","子" and etc. Even in this simple example, we can also see that there are 5^6 possible Chinese sequences which can be generated. It is easy to show that the number of the possible sequence is exponential to the length of source or target sequence.

Formulated as a sequence labeling task, PTC will require a much larger label set to work on than any other traditional sequence labeling tasks such as named entity recognition (NER) or part-of-speech (POS) tagging. In machine learning, sequence labeling is a type of pattern recognition task that involves the algorithmic assignment of a categorical label to each member of a sequence of observed values. Sequence labeling can be treated as a set of independent classification tasks,one per element of the sequence. Typically, the latter have dozens of labels, while the former will have thousands of ones. A too large label set makes the sequence labeling inefficient and low-performance.

In this paper, we propose a new approach by formulating the PTC problem as a machine translation task. Considering the obvious constraint that the target Chinese sequence's order keeps the same as the source Pinyins' order, there exists no reordering step in the translation procedure. It greatly alleviates the difficulty of training such a machine translation system. In this sense, this approach is similar to a monotone SMT, which means that we can decode the source sentence from left to right without any reordering. At the same time, we can also make a full use of the phrase-based features in the machine translation framework and effective parameter estimation method. The motivation for our works lie in the phenomenon that the whole sentence pinyin input method is far more mature and even for the typical input method, there are also many conversion errors which need people to correct manually, this way heavily reduces the efficiency of people's work efficiency.

The rest of the paper is organized as follows: Section 2 describes previous relevant works about PTC problem. Section 3 introduces the proposed approach. Experimental results are given in Section 4. Then a discussion about the experiment result are given in Section 5. We reach our conclusion in Section 6.

2 Related Work

Similar with the task of PTC, the grapheme-to-phoneme or phoneme-to-grapheme conversion problem has also developed many different approaches. For example, (Chen, 2003) introduces several models for grapheme-to-phoneme conversion, including a joint conditional maximum entropy model, a joint maximum n-gram model and a joint maximum n-gram model with syllabification.

To effectively solve the PTC problem, many natural language processing techniques have been applied. By and large, these methods can be separated into two main categories: rule-based methods and statistical methods. the rule-based methods can make use of concrete linguistic information to understand language meanwhile plentiful features and

automatic learning and prediction can be integrate to the statistical one effectively.

Wang et al. (2004) put forward a rough set approach to extract a number of rules from the corpus. (Zhang et al., 2006) presented an error correction post-processing approach based on grammatical and semantic rules. However, natural languages are so sophisticated that the rule-base methods can not effectively tackle all the situations. Recently, most works turn to statistical learning methods.

One of the earliest attempts to address this problem is to make use of language models. Now, many Chinese Pinyin input methods are still based on this model. (chen and Lee, 2000) successfully applied language models to the Chinese Pinyin input method. (Lee, 2003) extended language models further to disambiguate the Chinese homophone.

(Liu and Wang, 2002) built a machine learning approach to solve Chinese Pinyin-to-character for small memory application. Their approach lied on iterative new word identification and word frequency increasing that results in more accurate segmentation of Chinese character gradually. Their work can be applied to many small-memory platform such as Personal Digital Assistant(PDA) and etc.

(Zhao. and Sun, 1998) presented a word-self-made Chinese Phonetic-Character Conversion(CPCC) algorithm based on the Chinese Character Bigram which combined the advantages of CPCC based on Chinese character N-gram and advantages of CPCC based on Chinese word N-gram.

The paper (Zhang, 2007) presented a way to transform Chinese Pinyins to Chinese characters based on hybrid word lattice and study the related problems with hybrid language model and algorithms to solve the word lattice.

In the work of (Zhou et al., 2007), they utilized a segment-based hidden Markov model for Pinyin-to-Chinese conversion compared with the character based hidden Markov model.

(Lin and Zhang, 2008) presented a novel Chinese language model and studies their application in Chinese Pinyin-to-character conversion. Their model associate a word with supporting context including the frequent sets of the word's nearby phrases and the distances of phrases to the word.

Support vector machine(SVM) can also be used to deal with PTC problem as PTC can also be regarded as classifying the Pinyin to one of the Chinese characters. SVM replaces minimizing empirical risk in the traditional machine learning methods with minimizing the structure risk principle and shows a satisfied performance. (Jiang et al., 2007) put forward a PTC framework based on the SVM model. It effectively overcomes the drawback that language models cannot conveniently integrate rich features, and achieves a state-of-the-art accuracy of 92.94%.

As one of the most frequent tools to the classification and sequence labeling problem, Maximum Entropy(ME) model were also adopted to settle the PTC issue as in (Wang et al., 2006). A Class-based MEMM model is proposed to address the PTC conversion problem through exploitation of the pinyin constraints.

(Li et al., 2009) applied the conditional random field(CRF) model to the PTC problem in order to alleviate the label bias problem that usually occurs in the ME model (Andrew et al., 2001). (Li et al., 2009) made use of the constraint that one Pinyin can only map to limited number of Chinese characters thus greatly reducing the computation cost. However, their results show that CRF model does not outperform ME model(Li et al., 2009) and the CRF training will cost about approximately 200 days.

Artificial Immune Network based model is proposed to deal with the task of PTC conversion(Jiang and Pang, 2009). They propose an online learning approach the problems of sparse data and independent identical distribution.

The PTC problem can also be seen as one kind of machine transliteration which aims to generate a string in target language given a character string in source language. (Li et al., 2004) proposed a joint source-channel model to allow direct orthographical mapping between two different languages.

(Hatori and Suzuki, 2011) applied the phrase-based SMT model to predict Japanese Pronunciation, however, the differences between our work and theirs lie in a visual aspects. Both Japanese and Chinese adopt Chinese characters in their writing system, the work of (Hatori and Suzuki, 2011) was approximately a task to predict the pronunciation of a Chinese character, and ours is to predict a Chinese character sequence from a Pinyin(pronunciation) sequence. The task defined in this paper as discussed

in the above is a much more difficult disambiguation task than the one in (Hatori and Suzuki, 2011). That is, a Chinese character seldom has multiple pronunciations, but the same pronunciation may refer to quite a lot of Chinese characters, usually, dozens of characters.

3 PTC Conversion Model

In this section, we apply a monotone phrasal SMT-based approach to solve the PTC problem. The whole framework is illustrated in Figure 1. Firstly, we should prepare a sentence aligned corpus, then do the word alignment process. After this, we need to extract a translation table from the aligned corpus. Then we will use all of the features to train a translation model. The last process is decoding the source sentence.

3.1 Translation Model

Our SMT model is based on the discriminative learning framework which contains different real-valued features. In this model, F is a given foreign sentence F=$f_1, f_2, ..., f_J$, and needs to be translated into another sentence E=$e_1, e_2, ..., e_I$. The real-valued features are defined over F and E as $h_i(E, F)$. The score can be given by a log-linear formulation(Och and Ney, 2004) with respect to a series of weight parameters λ_1,...,λ_n. For a given source language sentence f, we can obtain the target language sentence e according to the following equation:

$$\begin{aligned} e_1^I &= \arg\max_{e_1^I} p_{\lambda_1^m}(e_1^I|f_1^J) \\ &= \arg\max_{e_1^I} \frac{exp[\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)]}{\sum_{\bar{e}_1^I} exp[\sum_{m=1}^{M} \lambda_m h_m(\bar{e}_1^I, f_1^J)]}, \end{aligned} \quad (1)$$

where h_m is the m-th feature function and λ_m is the m-th feature weight. The most common features used in modern phrased-based machine translation include phrase translation feature, language model feature, reordering model feature and word penalty feature.

As usual, to train the SMT model parameters, we adopt the minimum error rate training(MERT)(Och, 2003), which obtained the model towards getting the highest score corresponding to the concrete evaluation metric. For the sequence decoding, we use a stack decoder(Germann et al., 2001).

3.2 Features

The following real-valued features are adopted for learning, the bidirectional phrase translation probabilities, $p(\hat{e}|\hat{f})$ and $p(\hat{f}|\hat{e})$, the bidirectional lexical weighting $lex(\hat{e}|\hat{f})$ and $lex(\hat{f}|\hat{e})$, the target Chinese character n-gram probability, $p(\hat{e})$ and the phrase penalty. The estimation of these features requires a training corpus with source and target alignment at the character or word level.

The bidirectional conditional phrase translation probability contain much richer information than the one directional phrase translation probability. When translating the source phrase $\hat{f}$ into the target phrase $\hat{e}$, we take both $p(\hat{e}|\hat{f})$: the target phrase's probability given the source phrase, and $p(\hat{f}|\hat{e})$: the source phrase's probability given the target phrase. The bidirectional conditional phrase translation probabilities can be estimated by the relative frequency of the phrases extracted from the aligned corpus. Note that the phrase used is not a meaningful word combination any more, it just refers to a series of consequent characters. In practice, a model using both translation directions, with the proper weight setting, often outperforms a model that uses only one direction.

The lexical weighting feature is such a measurement that can be effectively used to estimate whether a phrase pair is reliable or not. Empirically, the lexical weighting(Berger et al., 1994; Brown et al., 1993; Brown et al., 1990) is defined as follows:

$$lex(e|f, a) = \prod_{i=1}^{length(e)} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall(i,j)\in a} w(e_i|f_j)$$

Here a is an alignment function defining each Chinese character with its corresponding Pinyin and w refers to the lexical conditional probability. The above equation shows that for the phrase pair(f,e), the translation probability can be interpreted as the product of the aligned lexical pairs (f_j,e_i). For the PTC conversion problem, the lexical pair refers to the pinyin-character pair. Based on the alignment we can estimate the possibility of the transformation of phrase pairs from the lexical translation aspect.

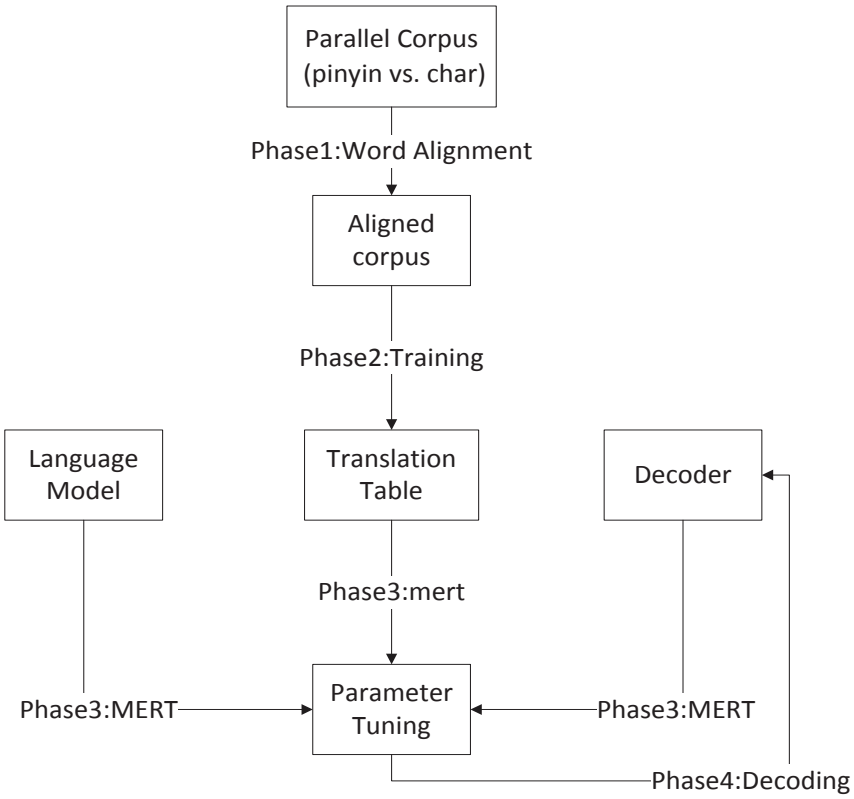

Figure 1: An overview of machine translation system which consists of several phases.

	Training			Development	Test
#sentence	10K	100K	1M	2K	100K
#character	452056	4371102	43679593	83765	4123184

Table 2: The size of different datasets

qi_气 wen_温 ye_也 zhou_骤 jiang_降 dao_到 17_17 she_摄 shi_氏 du_度 ._。

features	samples
pinyin	zhou
suffix	hou,ou,u
prefix	z,zh,zho
previous pinyin	ye
previous character	也
pre-pre pinyin	wen
previous two pinyins	wenye
next pinyin	jiang
next next pinyin	dao

Figure 2: The sample training sentence and its ME features.

The phrase penalty is used to estimate the preference towards a sentence which has more segmented phrases or less segmented phrases. Practically, a factor is introduced for each phrase translation. If the factor is less than 1 we would prefer a longer phrase and otherwise shorter phrase is preferred.

4 Experiment

It is natural to formulize PTC as a sequence labeling task, which usually adopt maximum entropy Markov model(Berger et al., 1996) as the standard tool in most existing literatures[2]. Thus we conducted a group of experiments to evaluate the proposed SMT approach with the ME model as the baseline system. The features we use are the most frequent used ones in related works(Wang et al., 2006; Li et al., 2009).

4.1 Experiment settings

Firstly, we realize a way to get large Pinyin and Chinese character sentence pairs because to our best knowledge there is no such open dataset available. Given a Chinese character sequence, it is much easier to convert it to a Pinyin sequence because when a Chinese character is put in a context, it usually has an unique Pinyin counterpart. Based on this observation, we label the Chinese text with Pinyins through the forward maximal matching algorithm (kwong Wong and Chan, 1996) incorporated with a word-Pinyin dictionary from Sogou [3]. The data from the People's Daily of 1998 year is used as the training set and the development and test data are token from 1997 year's. The size of datasets is in table 2, the data of 10K and 100K are extracted from the data of 1M. Then we check the auto-labeled data and correct few mistakes.

The sample sentence "qi_气 wei_温 ye_也 zhou_骤 jiang_降 dao_到 ling_零 xia_下 17_1 7 she_摄 shi_氏 du_度 ._。(The temperature also dropped abruptly to seventeen below zero centidegrees.)" is shown in figure2, where the Pinyin and the Chinese character is separated by "_".

4.2 Maximum Entropy model

The implementation of ME model is from the OpenNLP tools[4].

4.2.1 Feature template

We assume the current Pinyin sequence is $p_1, ..., p_n$ and the corresponding Chinese character sequence is $c_1, ..., c_n$. The current Pinyin is p_k. As usually being regarded as an sequence labeling task, we design the feature set for the ME model as follows:

- the current Pinyin itself p_k;
- the suffixes of the Pinyin. For a given Pinyin s which is made of $s_1, ..., s_n$, the suffix of s refers to the substrings $s_i, ..., s_n (i \geq 2)$;
- the prefixes of the Pinyin. For a given Pinyin s which is made of $s_1, ..., s_n$, the suffix of s refers to the substrings $s_1, ..., s_i (i < 2)$;
- the previous Pinyin p_{k-1};
- the Chinese character c_{k-1} with respect to the previous Pinyin p_{k-1}(Markov feature);
- the Pinyin before previous Pinyin p_{k-2};
- the Pinyin before previous Pinyin and the previous Pinyin $p_{k-2}p_{k-1}$;
- the next Pinyin p_{k+1};

[2]Though conditional random field has shown more effective than ME model to solve sequence labeling problem, it is not a practical tool for PTC due to too many labels that PTC requires causing too high computational cost.

[3]The resource includes 4,083,906 Chinese word and Pinyin pairs, and it can be download from http://code.google.com/p/hslinuxextra/downloads/list.

[4]The tool can be downloaded from http://incubator.apache.org/opennlp/index.html

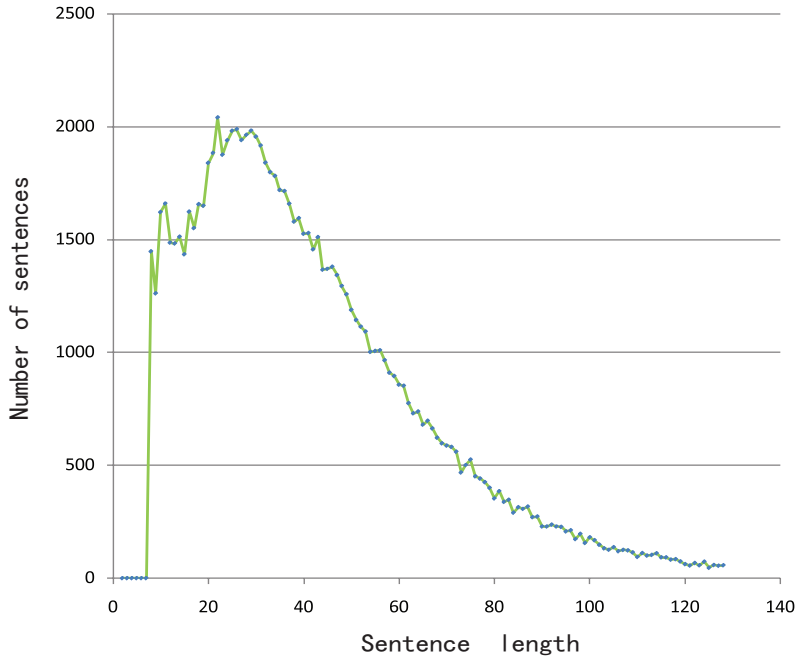

Figure 3: The sentences' length distribution.

Model \ Dataset	10K	100K	1M
ME	0.829	0.891	0.933
SMT	0.947	0.952	0.955

Table 3: The accuracy for ME model and SMT model on different datasets in terms of words.

- the Pinyin after the next one p_{k+2};

Figure 2 illustrates a full feature set sample, for the given sample sentence at the upper part of the figure, all related features for Pinyin-character pair, "zhou_骤(abruptly)", can be shown in the bottom table of the Figure.

Finally, the converted Chinese character sequences are compared to the golden data, the accuracy results can be seen in Table 3.

4.3 Machine Translation Framework

In this experiment, we conduct the process based on Stanford's phrasal(Cer et al., 2010) which is an open source phrase-based machine translation system. For the traditional phrase-based machine translation method, the processing steps are often stated as following:

- train an alignment model from the parallel corpus(not needed for our experiments.)
- extract phrases based on the former alignment model
- minimum error rate training
- decoding

As we have known that it must be an one-to-one alignment for PTC, it is unnecessary to train the alignment model and the phrases can be directly extracted based on the one-to-one alignment of character and Pinyin. Our experiment is based on 3-gram language model and our maximum phrase length is set to 7.

The results given by the SMT approach are in Table 3. We get the results based on three different training sets.

5 Discussion

In this section, we make a detailed experimental analysis to distinguish the result of the SMT model from that of the ME model on the whole sentence accuracy and time cost.

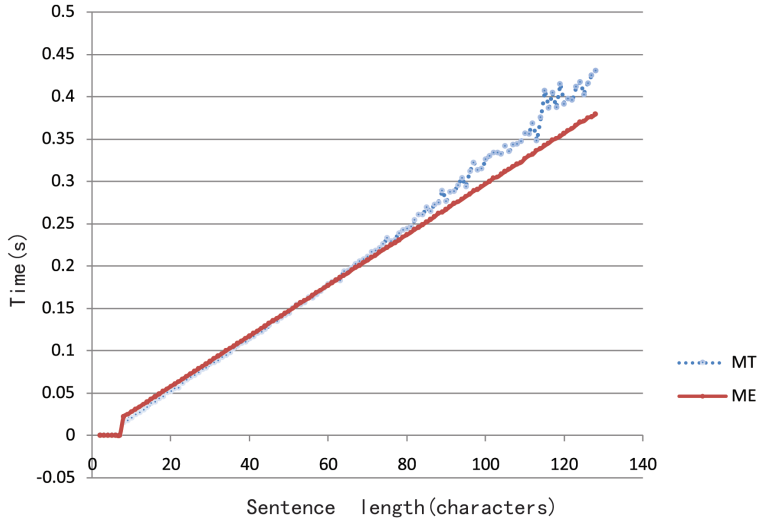

Figure 4: The comparison of decoding time of SMT model and ME model.

5.1 Main Result

Table 3 shows the main results of our experiments. Here the accuracy means the percentage of the correct labels in our decoding results. On all of three training data sets, the results of SMT are much better than ME.

To illustrate this point concretely, we can see the different result produced by the two models on the sample sentence

- Pinyin Sequence: *zhe yi cheng ji zai quan guo wu da tie lu ju zhong ming lie bang shou*
- Character sequence: *这一成绩在全国五大铁路局中名列榜首(This result ranks the best among the five biggest railway bureaus all over the country)*

ME model outputs "这一成绩在全国五大铁路局中名列帮手" while the result of SMT model is "这一成绩在全国五大铁路局中名列榜首". The ME model makes an error as it translate 'bang shou' into '帮手'(helper) and the SMT model outputs are the completely equal to the golden sentence, and 'bang shou' has been correctly translated into '榜首'(the best on the list).

By comparing outputs of these two sentence we can see that the SMT model is much more representative than the ME model. As the features we defined are based on the phrase pairs, the model can deduce that the score of target sentence which is composed of phrase pair(ming lie bang shou,名列榜首(who is best on the list)) is greater than the score of sentence which is compose with phrase pair(ming lie,名列(who is)) and phrase pair(bang shou, 帮手(helper)). This result also verifies the effectiveness of the SMT features to capture the local property of the source sentence and the target sentence and can combine longer dependencies.

To show how the proposed SMT approach outperforms the ME model, we give a comparison on another metric, the whole sentence accuracy which represents the ratio how many sentences are completely correctly decoded by the system. This metric could be very useful to evaluate a practical Chinese input method. As even one incorrect decoded character may ask human users to pay too many keyboard hits to correct, which user has to backspace the cursor one by one and re-choose the right character candidate one by one, the whole sentence accuracy could be more effective to evaluate user experience of a Chinese input method. Besides, the whole sentence's accuracy also reflects the model's efficiency in another view.

The distribution of sentence length is shown in Figure 3, from which we can see that most sentences are of length between 20 Chinese characters and 40 Chinese characters. The whole sentence's accuracy for both these two models can be shown in Table 4. From this table we can see that the results of SMT model is much better than the ME model, which in-

Model \ Dataset	10K	100K	1M
ME	0.075	0.169	0.302
SMT	0.402	0.429	0.454

Table 4: The whole sentence accuracy on test dataset.

dicates that a SMT decoder for PTC could bring out much better user experience.

5.2 Time Cost

For the training time of these two models, we make a comparison on the biggest training dataset, which has 1M training sentences. It took about a week or so to train a ME model while the training time of our approach cost about within one day which is much faster than the that of ME model. From the description in (Li et al., 2009) we know that the training of CRF would cost much more than ME and the result of CRF is not better than ME.

Being a core component of Chinese input method, PTC is sensitive to the computational cost. Thus time cost of decoding for the two models is reported as follows.

To make the differences more exactly, the decoding time of the two models is compared on sentences with the same length. The results are shown in Figure 4. We can see that the decoding time increases when the sentence length becomes larger. However, even when the sentence length is larger than 120 characters, the decoding time is still less than 0.45s. From this graph, it is apparent that the ME model decoding is slightly faster than the SMT model as the sentence is quite long. However, for most sentences with 20 to 40 characters, the SMT model does not decodes slower than the ME model.

6 Conclusion

We present a novel approach to the problem of Pinyin-to-character conversion(PTC). Motivated by the similarities between machine translation and PTC, we re-formulize the latter as a simplified machine translation problem. In the new formulization, the most computational expensive part of machine translation, alignment learning, could be conveniently ignored by considering that PTC could build one-to-one mapping pairs in the whole text. Meanwhile, the SMT model for PTC maintains the merit that it integrates more effectively helpful features to outperform the baseline system, ME model, which is a standard sequence labeling tool for traditional PTC task. A group of experiments are carried out to verify the effective of the proposed MT model. The results show that MT model outperforms the previous ME model and provides satisfactory performance.

References

McCallum Andrew, Pereira Fernando, and Lafferty John. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, Williamstown, MA.

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. GiUet, John D. Lafferty, Robert L. Mercer, Harry Printz, and Luboš Ureš. 1994. The candide system for machine translation. In *Proceedings of the workshop on Human Language Technology*, pages 157–162. Association for Computational Linguistics.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D.Manning. 2010. Phrasal: A statistical machine translation toolkit for exploring new model features. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 9–12, Los Angeles, California, June. Association for Computational Linguistics.

Zheng Chen and Kai-Fu Lee. 2000. A new statistical approach to chinese pinyin input. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 241–247, Hong Kong. Association for Computational Linguistics.

Stanley F. Chen. 2003. Conditional and joint models for grapheme-to-phoneme conversion. In *Eighth European Conference on Speech Communication and Technology*.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel marcu, and Kenji Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 228–235. Association for Computational Linguistics.

Jun Hatori and Hisami Suzuki. 2011. Japanese pronunciation prediction as phrasal statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 120–128. Asian Federation of Natural Language Processing.

Wei Jiang and Xiuli Pang. 2009. An artificial immune network approach for pinyin-to-character conversion. In *Virtual Environments, Human-Computer Interfaces and Measurements Systems, 2009. VECIMS'09. IEEE International Conference on*, pages 27–32. IEEE.

Wei Jiang, Yi Guan, Xiaolong Wang, and BingQuan Liu. 2007. Pinyin-to-character conversion model based on support vector machines. *Journal of Chinese information processing*, 21(2):100–105.

Pak kwong Wong and Chorkin Chan. 1996. Chinese word segmentation based on maximum matching and word binding force. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 200–203. Association for Computational Linguistics.

Yue-Shi Lee. 2003. Task adaptation in stochastic language model for chinese homophone disambiguation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(1):49–62.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 159–166. Association for Computational Linguistics.

Lu Li, Xuan Wang, Xiaolong Wang, and Yanbing Yu. 2009. A conditional random fields approach to chinese pinyin-to-character conversion. *Journal of Communication and Computer*, 6(4):25–31.

Bo Lin and Jun Zhang. 2008. A novel statistical chinese language model and its application in pinyin-to-character conversion. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1433–1434. ACM.

Bingquan Liu and Xaiolong Wang. 2002. An approach to machine learning of chinese pinyin-to-character conversion for small-memory application. In *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, volume 3, pages 1287–1291. IEEE.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Xiaolong Wang, Qingcai Chen, and Daniel So Yeung. 2004. Mining pinyin-to-character conversion rules from large-scale corpus: a rough set approach. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(2):834–844.

Xuan Wang, Lu Li, Lin Yao, and Waqas Wanwar. 2006. A maximum entropy approach to chinese pin yin-to-character conversion. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 2956–2959. IEEE.

Yan Zhang, Bo Xu, and Chengqing Zong. 2006. Rule-based post-processing of pinyin to chinese characters conversion system. In *International Symposium on Chinese Spoken Language Processing*.

Sen Zhang. 2007. Solving the pinyin-to-chinese-character conversion problem based on hybrid word lattice. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, 30(7):1145–1153.

Yibao Zhao. and Shenghe Sun. 1998. A word-self-made chinese phonetic-character conversion algorithm based on chinese character bigram [j]. *ACTA ELECTRONICA SINICA*, 10.

Xiaohua Zhou, Xiaohua Hu, Xiaodan Zhang, and Xiajiong Shen. 2007. A segment-based hidden markov model for real-setting pinyin-to-chinese conversion. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 1027–1030. ACM.