

# Neural-based Pinyin-to-Character Conversion with Adaptive Vocabulary

Yafang Huang<sup>1,2</sup>, Zhuosheng Zhang<sup>1,2</sup>, Hai Zhao<sup>1,2</sup>,

<sup>1</sup>Department of Computer Science and Engineering,

Shanghai Jiao Tong University, Shanghai, 200240, China

<sup>2</sup>Key Laboratory of Shanghai Education Commission for Intelligent Interaction  
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

huangyafang@sjtu.edu.cn, zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

## Abstract

Pinyin-to-character (P2C) conversion is the core component of pinyin-based Chinese input method engine (IME). However, the conversion is seriously compromised by the ambiguities of Chinese characters corresponding to pinyin as well as the predefined fixed vocabularies. To alleviate such inconveniences, we propose a neural P2C conversion model augmented by a large online updating vocabulary with a target vocabulary sampling mechanism. Our experiments show that the proposed approach reduces the decoding time on CPUs up to 50% on P2C tasks at the same or only negligible change in conversion accuracy, and the online updated vocabulary indeed helps our IME effectively follows user inputting behavior.

## Introduction

Chinese may use more than twenty thousand different Chinese characters so that it is non-trivial to type the Chinese character directly from a Latin-style keyboard which only has 26 keys. The pinyin as the official romanization representation for Chinese provides a solution that maps the Chinese character to a string of Latin alphabets so that every character has a letter writing form of their own and users can type pinyin in terms of Latin letters to input Chinese characters into computer. Therefore, converting pinyin to Chinese characters is the most basic module of all pinyin-based IMEs.

As each Chinese character may be mapped to a pinyin syllable, it is natural to regard the Pinyin-to-Character (P2C) conversion as a machine translation between two different languages, pinyin sequences and Chinese character sequences (namely Chinese sentence). Actually, such a translation in P2C procedure is even more straightforward and simple by considering that the target Chinese character sequence keeps the same order as the source pinyin sequence, which means that we can decode the target sentence from left to right without any reordering.

Meanwhile there exists an obvious challenge in P2C procedure, unlike English and other alphabetic languages whose words have one-to-one mapping between words and combination of letters, there are only about 500 pinyin syllables corresponding to ten thousands of Chinese characters,

even though the amount of the commonest is more than 6,000 (Jia and Zhao 2014; Zhang et al. 2014). As well known, the homophone and the polyphone is quite common in Chinese language. Actually, one pinyin may correspond to ten or more Chinese characters on the average.

However, pinyin IME may benefit from decoding longer pinyin sequence for more efficient inputting. When a given pinyin sequence becomes longer, the list of the corresponding legal character sequences will significantly reduce. For example, IME being aware of that pinyin sequence “bei\_jing” can be only converted to either “背景(background)” or “北京(Beijing)” will greatly help it make the right and more efficient P2C decoding, as both pinyins “bei” and “jing” are respectively mapped to dozens of difference single Chinese characters. Table 1 illustrates that the list size of the corresponding Chinese character sequence converted by pinyin sequence “bei jing huan ying ni” (北京欢迎你, Welcome to Beijing) is changed according to the different sized source pinyin sequences.

Pinyin seq. consists of <b>1</b> syllable	bei	jing	huan	ying	ni
	被	敬	环	英	你
	北	静	换	颖	昵
	呗	井	还	迎	逆
	杯	京	幻	影	拟
	背	经	欢	应	尼
Pinyin seq. consists of <b>2</b> syllables	bei_jing		huan_ying		ni
	北京		幻影		你
	背景		欢迎		你
Pinyin seq. consists of <b>5</b> syllables	bei_jing_huan_ying_ni				
	北京欢迎你				

Table 1: The shorter the pinyin sequence is, the more character sequences will be mapped.

To reduce the P2C ambiguities by decoding longer input pinyin sequence, Chinese IMEs may often utilize word-based language models since character-based language model always suffers from the mapping ambiguity issues.

However, the effect of the work in P2C will be undermined with quite restricted vocabularies. The efficiency of IME conversion depends on the size of the vocabulary and previous work on machine translation has shown a large vocabulary is necessary to achieve good accuracy (Jean et al. 2015). What's more, some sampling techniques for vocabulary selection are proposed which do not leverage the conversion cost (Zhou et al. 2016; Wu et al. 2018). As IMEs work, users inputting style may change from time to time, let alone diverse user may input quite diverse contents, which makes a predefined fixed vocabulary can never be sufficient. For a convenient solution, most commercial IMEs have to manually update their vocabulary on schedule. Moreover, the training for word-based language model is especially difficult for rare words, which appear sparsely in the corpus but generally take up a large share of the dictionary.

In this paper, we first introduce an online sequence-to-sequence (seq2seq) model for P2C, and then design a sampling mechanism utilizing our online updated vocabulary to enhance the conversion accuracy of IMEs as well as speed up the decoding procedure. What's more, a character-enhanced word embedding (CWE) mechanism is proposed to represent the word. In detail, the proposed model can let IME work generally at the word level and pick a very small target vocabulary for each sentence. Moreover, every time user makes a selection contradicted the prediction given by the P2C conversion module, the module will update the vocabulary accordingly. Our evaluation will be performed on three diverse corpora, including two that is from real user inputting history in recent years, for verifying the effectiveness of the proposed method in different scenarios.

The rest of the paper is organized as follows: Section 2 discusses relevant works. Sections 3 and 4 introduces the proposed model. Experimental results and the model analysis are in Sections 5 and 6. Section 7 concludes this paper.

## Related Work

To effectively utilize words for IMEs, many natural language processing (NLP) techniques have been applied. (Chen 2003) introduced a joint maximum n-gram model with syllabification for grapheme-to-phoneme conversion. (Chen and Lee 2000) used a trigram language model and incorporated word segmentation to convert pinyin sequence to Chinese word sequences. (Luo and Roukos 2008) proposed an iterative algorithm to discover unseen words in corpus for building Chinese language model. (Mori, Takuma, and Kurata 2006) described a method enlarging the vocabulary which can capture the context information. However, all the above methods require an predefined fixed vocabulary.

For either pinyin-to-character for Chinese IMEs or kana-to-kanji for Japanese IMEs, a few language model training methods have been developed. (Mori et al. 1998) proposed probabilistic based language model for IME. (Jiampo-jamarn, Cherry, and Kondrak 2008) presented an online discriminative training. (Lin and Zhang 2008) proposed a statistic model using the frequent nearby set of the target word. (Chen, Wu, and He 2012) used collocations and k-means clustering to improve the n-pos model for Japanese IME. (Jiang et al. 2007) put forward a PTC framework based

on support vector machine. (Hatori and Suzuki 2011) and (Yang, Zhao, and Lu 2012) respectively applied statistic machine translation (SMT) to Japanese pronunciation prediction and Chinese P2C tasks. (Chen, Wang, and Zhao 2015; Huang et al. 2018) regarded the P2C as a translation between two languages and solved it in neural machine translation framework.

All the above mentioned work, however, still rely on a predefined fixed vocabulary, and IME users have no chance to refine their own dictionary online. (Zhang, Wei, and Zhao 2017) is mostly related to this work, who also offers an online mechanism to adaptively update user vocabulary. The key difference between these two work lies on that this work presents the first neural solution with online vocabulary adaptation as for our best knowledge.

The efficiency of neural models depends on the size of the target vocabulary and previous work has shown that vocabularies of well over 50k word types are necessary to achieve good accuracy (Jean et al. 2015) (Zhou et al. 2016). Neural translation systems compute the probability of the next target word given both the previously generated target words as well as the source sentence. Estimating this conditional distribution is linear in the size of the target vocabulary which can be very large for many language pairs. Recent work in neural translation has adopted vocabulary selection techniques from language modeling which do not leverage the input sentence (Mi, Wang, and Ittycheriah 2016; ?; ?).

A recent line of studies can prove the demonstrable effects of word representation, such as language modeling (Verwimp et al. 2017) and machine translation (Choi, Cho, and Bengio 2017). As for improved word representation in IMEs, (Hatori and Suzuki 2011) solved Japanese pronunciation inference combining word-based and character-based features within SMT-style framework to handle unknown words. (Neubig et al. 2013) proposed character-based SMT to handle sparsity. (Okuno and Mori 2012) introduced an ensemble model of word-based and character-based models for Japanese and Chinese IMEs. All the above mentioned methods used similar solution about character representation for various tasks.

Our work takes inspiration from (Luong and Manning 2016), (Mi, Wang, and Ittycheriah 2016) and (Cai et al. 2017). The former built a novel representation method to tackle the rare word for machine translation. In detail, they used word representation network with characters as the basic input units. For each sentence or batch, the second paper proposed a method which only predicts the target words in its sentence-level or batch-level vocabulary. Cai et al.(2016) presented a greedy neural word segmenter with balanced word and character embedding inputs. High-frequency word embeddings are attached to character embedding via average pooling while low-frequency words are computed from character embedding. Our embeddings also contains different granularity level of embedding, but the word vocabulary is capable of being updated in accordance of users' inputting choice during IME working. In contrast, (Cai et al. 2017) build embeddings based on the word frequency of a fixed corpus.

## Our Models

For a convenient reference, hereafter a *character* in pinyin language also refers to an independent syllable in the case without causing confusion, and *word* means a pinyin syllable sequence.

As illustrated in Figure 1, the core of our hybrid P2C is based on the seq2seq model (Cho et al. 2014) in terms of encoder-decoder framework. Given a pinyin sequence  $X$  and a Chinese character sequence  $Y$ , the encoder of our neural P2C model utilizes a network for pinyin representation in which both word-level and character-level embedding are exploited, and the decoder is to generate the target Chinese sequence which maximizes  $P(Y|X)$  using maximum likelihood training.

Starting from an initial vocabulary with indicator from each turn of user inputting choice, the online learning module helps to update the word vocabulary by minimizing empirical prediction risk.

### Pinyin-Character Parallel Corpus

Pinyin-character parallel corpus can be conveniently generated by automatically annotating Chinese character text with pinyin as introduced in (Yang, Zhao, and Lu 2012). Using standard Chinese word segmentation methods, we may segment both character and pinyin text into words with the same segmentation for each sentence pair.

### Encoder-Decoder

The encoder is a bi-directional long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997). The vectorized inputs are fed to forward and backward LSTMs to obtain the internal representation of two directions. The output for each input is the concatenation of the two vectors from both directions. Our decoder is based on the global attentional models proposed by (Luong, Pham, and Manning 2015) to consider the hidden states of the encoder when deriving the context vector. The probability is conditioned on a distinct context vector for each target word. The context vector is computed as a weighted sum of previous hidden states. The probability of each candidate word as being the recommended one is predicted using a softmax layer over the inner-product between source embeddings and candidate target characters. Figure 2 shows the architecture.

### Online P2C Learning with Vocabulary Adaptation

As the core of Chinese IME, P2C conversion, has been formulated into a seq2seq model as neural machine translation (NMT) between pinyin and character sequences, there are still a few differences between the P2C and standard machine translation. 1) Considering both pinyin syllables and Chinese characters are segmented into single-character *word* as Figure 3a, there is a one-to-one mapping between any character and its pinyin syllable without word reordering, while typical machine translation does not enjoy such benefits and has to perform careful word reordering explicitly or implicitly. 2) As Chinese language is always sensitive to

the segmentation scheme, in the writing of either the Chinese character or the pinyin, P2C as NMT may suffers from alignment mismatch on both sides like Figure 3b or benefit a lot from perfect one-to-one alignment like Figure 3c, while typical machine translation is seldom affected by such segmentation alignment. 3) P2C as a working component of IME, every time it returns a list of Chinese character sequence predictions, user may indicate which one is what he or she actually expects to input. To speed up the inputting, IME always tries to rank the user’s intention at the top-1 position. So does IME, we say there is a correct conversion or prediction. Different from machine translation job, users’ inputting choice will always indicate the ‘correct’ prediction right after IME returns the list of its P2C conversion results.

Therefore, IME working mode implies an online property, we will let our neural P2C model also work and evaluate in such a way. Meanwhile, online working means that our model has to track the continuous change of users’ inputting contents, which is equally a task about finding new words in either pinyin sequence or character sequence.

However, word should be obtained through the operation of word segmentation. Note that as we have discussed above, segmentation over pinyin sequence is also necessary to alleviate the ambiguity of pinyin-to-character mapping. Thus the task here for IME online working actually requires an online word segmentation algorithm. Our solution to this requirement is adopting a vocabulary-based segmentation approach, namely, the maximum matching algorithm which greedily segments a longest matching in the given vocabulary at the current segmentation point of a given sequence. Then adaptivity of the segmentation thus actually relies on the vocabulary online updating.

Algorithm 1 gives our online vocabulary updating algorithm for one-turn IME inputting. Note the algorithm maintains a pinyin-character bilingual vocabulary. Collecting user’s inputting choices through IME, our P2C model will perform online training over segmented pinyin and character sequences with the updated vocabulary. The updating procedure introduces new words by comparing user’s choice and IME’s top-1 prediction. The longest mismatch  $n$ -gram characters will be added as new word.

We adopt a hybrid mechanism to balances both words and characters representation, namely, Character-enhanced Word Embedding (CWE). At the beginning, we keep an initial vocabulary with the most frequent words. The words inside the vocabulary are represented as enhanced-embedding, and those outside the list are computed from character embeddings. A pre-trained word2vec model (Mikolov et al. 2013) is generated to represent the word embedding  $WE(w) (w \in \hat{V})$ . At the same time we feed all characters of each word to a bi-gated recurrent unit (bi-GRU) (Cho et al. 2014) to compose the character level representation  $CE(w) (w = \{c_i | i = 1, 2, 3, \dots\})$ .

The enhanced embedding  $CWE(w)$  is to straightforwardly integrate word embedding  $WE(w)$  and character embedding  $CE(w)$  by element-wise multiplication.

$$CWE(w) = WE(w) \odot CE(w)$$

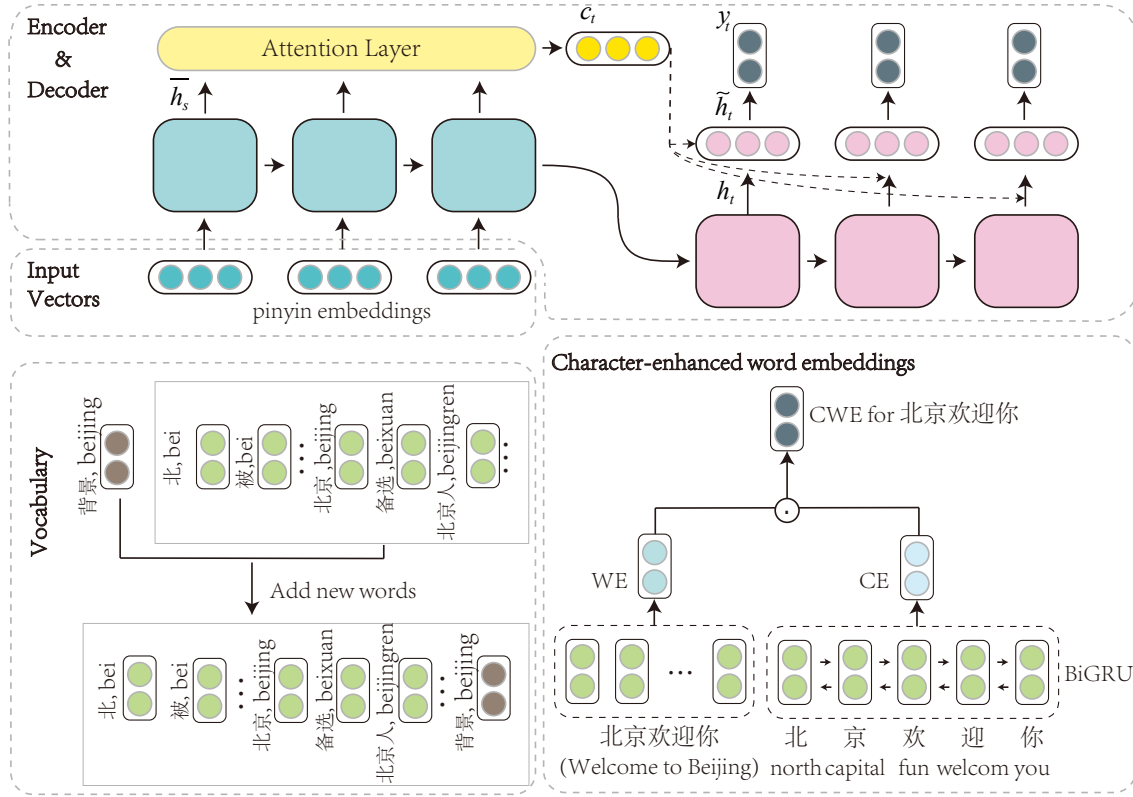


Figure 1: Architecture of the proposed Neural-based Chinese Input Method

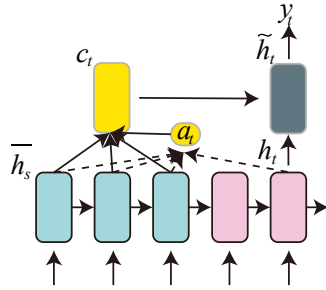


Figure 2: Architecture of the attention-based encoder-decoder model.

- (a) bei | jing | huan | ying | ni  
北 | 京 | 欢 | 迎 | 你
- (b) bei | jing | huan ying | ni  
北 京 | 欢 | 迎 | 你
- (c) bei jing | huan ying | ni  
北 京 | 欢 迎 | 你

Figure 3: Different segmentations decide different alignments

### Target Vocabulary Selection

In this section, we aim to use a large vocabulary of  $V$  and reduce the size of target vocabulary  $\hat{V}$  as small as possible

to reduce the computing time. Our basic idea is to maintain a separate and small vocabulary  $\hat{V}$  for each sentence so that we only need to compute the probability distribution over a small vocabulary for each sentence.

We first generate a sentence-level vocabulary  $V_s$  to be one part of our  $\hat{V}$ , which including the mapped Chinese words of each pinyin in the source sentence. As bilingual vocabulary  $V$  consists of the pinyin and Chinese word pair of all the words that have appeared, it's naturally to use a prefix maximum matching algorithm give a sorted list of relevant candidate translations  $D(x) = [Ch_1, Ch_2, \dots]$  for the source pinyin. Thus, we generate a target vocabulary  $V_s$  for a sentence  $x = (Py_1, Py_2, \dots)$  by merging all the candidates of all pinyins.

In order to cover target un-aligned functional words, we need top  $n$  most common target words  $V_c$ .

In training procedure, the target vocabulary  $\hat{V}$  for a sentence  $x$  need to include the target words  $V_t$  in the reference  $y$ .

$$\hat{V} = V_s \cup V_c \cup V_y$$

In decoding procedure, the  $\hat{V}$  only contains two parts,

$$\hat{V} = V_s \cup V_c$$

---

**Algorithm 1** Online Vocabulary Updating Algorithm

---

**Input:**

- Vocabulary:  $V = \{(Py_i, Ch_i) | i = 1, 2, 3, \dots\}$ ;
- Input pinyin sequence:  $Py = \{py_i | i = 1, 2, 3, \dots\}$ ;
- IME predicted top-1 character sequence:  $Cm = \{cm_i | i = 1, 2, 3, \dots\}$ ;
- User choosing character sequence:  $Cu = \{cu_i | i = 1, 2, 3, \dots\}$ .

**Output:**

- The Updated Vocabulary:  $\hat{V}$ .

```
1: ▷ Adding new words
2: for  $n = 6$  to  $2$  do
3:   Compare n-gram of  $Cu$  and  $Cm$ 
4:   if Mismatch  $Ch$  is found // both the first and last characters are different at least then
5:     if  $Ch$  is not in  $\hat{V}$  then
6:        $V = V \cup \{Ch\}$ 
7:     end if
8:   end if
9:   if no mismatch is found then
10:    break
11:  end if
12: end for
13: return  $\hat{V}$ ;
```

---

## Experiment

### Datasets and Evaluation Metrics

We adopt two corpora for evaluation. The People’s Daily corpus is extracted from the People’s Daily from 1992 to 1998 by Peking University (Emerson 2005). The bilingual corpus can be straightforwardly produced by the conversion proposed by (Yang, Zhao, and Lu 2012). Contrast to the style of the People’s Daily, the TouchPal corpus (Zhang, Wei, and Zhao 2017) is a large scale of user chat history collected by TouchPal IME, which are more colloquial. Hence, we use last two corpora to simulate user’s chatting input to verify our online model’s adaptability to different environments. The test set size is 2,000 in both corpora.

The statistics of two corpora are shown in Table 2<sup>1</sup>.

Two metrics are used for our evaluation by following previous work: Maximum Input Unit (MIU) Accuracy (Zhang, Wei, and Zhao 2017) and KeyStroke Score (KySS) (Jia and Zhao 2013). The former measures the conversion accuracy of MIU, whose definition is the longest uninterrupted Chinese character sequence inside a sentence. As the P2C conversion aims to output a rank list of corresponding character sequences candidates, the top- $K$  MIU accuracy means the possibility of hitting the target in the first  $K$  predict

<sup>1</sup>As no publicly released corpus for training and evaluation of IME systems, we will also release the two corpora as this paper is published.

items. We will follow the definition of (Zhang, Wei, and Zhao 2017) about top- $K$  accuracy. The KySS quantifies user experience by using keystroke count. An IME with higher KySS is supposed to perform better. For an ideal IME, there will be  $KySS = 1$ .

### Settings

IME works giving a list of character sequence candidates for user choosing. Therefore, measuring IME performance is equivalent to evaluating such a rank list. In this task, we select 5 converted character sequence candidates for each pinyin sequence. Given a pinyin sequence and candidate characters, our model is designed to rank the characters in an appropriate order.

Here are the model setting we used: a) pre-trained word embeddings were generated on the People’s Daily corpus; b) the RNN for encoder and decoder have 3 layers and 500 cells, and the representation networks have 1 layer; c) the initial learning rate is 1.0 and we will halve the learning rate every epoch after 9 epochs; d) dropout is 0.3; e) the default frequency filter ratio for CWE establishment is 0.9; f) the  $\beta$  for calculating rank score is 0.95. The same setting is applied to all models.

For a balanced treatment over both corpora, we used baseSeg (Zhao et al. 2006) to segment all text, then extract all resulted words into the initial vocabulary for late online evaluation. We train the base P2C models for 13 epochs with plain stochastic gradient descent on the People’s Daily corpus with 32 batch size, and the online training process runs 25 epochs with 1 batch size. In practical application, we perform online training for once every 64 instances are inputs to control costs.

### Results

We compare our P2C conversion system with two baseline systems, Google IME <sup>2</sup> and Offline and Online Model for

<sup>2</sup>The Google IME is the only commercial Chinese IME providing a debuggable API on the market now.

		Chinese	Pinyin
PD	# MIUs	5.04M	
	# Word	24.7M	24.7M
	# Vocab	54.3K	41.1K
	# Target Vocab (train)	2309	-
	# Target Vocab (dec)	2168	-
TP	# MIUs	689.6K	
	# Word	4.1M	4.1M
	# Vocab	27.7K	20.2K
	# Target Vocab (train)	2020	-
	# Target Vocab (dec)	2009	-

Table 2: MIUs count, word count and vocab size statistics of our training data. PD refers to the People’s Daily, TP is TouchPal corpus.

System	ED	PD			TP		
		Top1	Top5	Top10	Top1	Top5	Top10
Existing P2C							
Google IME		70.9	78.3	<b>82.3</b>	57.5	63.8	69.3
OMWA		55.0	63.7	70.2	19.7	24.8	27.7
On-OMWA		64.4	72.9	77.9	57.1	71.1	80.9
Our P2C							
Base P2C	200	53.2	64.7	70.3	46.8	68.8	75.7
On-P2C	200	68.1	77.3	78.2	69.8	88.7	89.3
On-P2C (bi)	200	70.5	79.8	80.1	71.0	89.2	89.5
On-P2C (bi)	300	70.8	80.5	81.2	<b>71.9</b>	89.6	<b>90.6</b>
On-P2C (bi)	400	<b>71.3</b>	80.1	<b>81.3</b>	71.7	<b>89.7</b>	90.3
On-P2C (bi)	500	69.9	78.2	81.0	70.7	89.2	89.8

Table 3: Top- $K$  accuracies on the People’s Daily (PD) , TouchPal (TP) corpora. ED refers to embedding dimension. The best results are in bold.

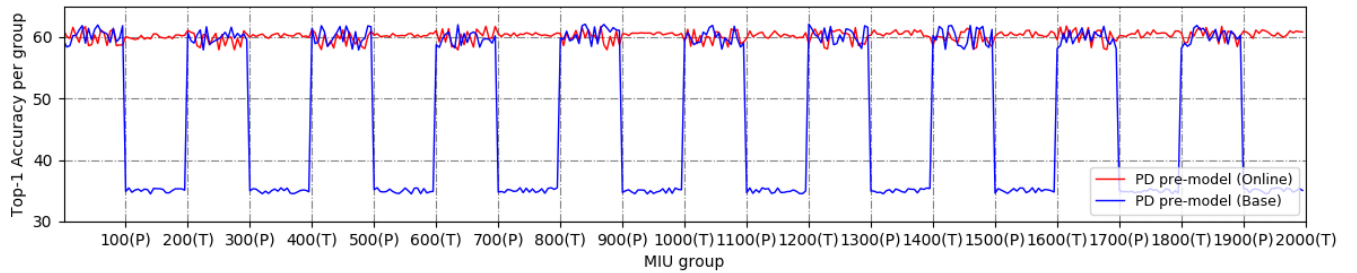


Figure 4: Top-1 accuracy on an interlaced joint Corpus. P: the People’s daily segment, T: Touchpal segment.

Filter Ratio	0	0.3	0.6	0.9	1.0
Top-5 Accuracy(valid set)	66.4	68.3	84.3	89.7	87.5
Top-5 Accuracy(test set)	66.3	68.1	83.9	<b>89.6</b>	87.1

Table 4: Top-5 accuracies of P2C after filtering specific ratio of words from vocabulary.

Models	the People’s Daily	TouchPal
Google IME	0.7535	0.6465
OMWA	0.6496	0.4489
On-OMWA	0.7115	0.7226
Base P2C	0.6922	0.7910
On-P2C	<b>0.8301</b>	<b>0.8962</b>

Table 5: User experience in terms of KySS

Word Acquisition (OMWA, On-OMWA)(Zhang, Wei, and Zhao 2017), and the results are shown in Table 3.

On the People’s Daily corpus, our online model (On-P2C) outperforms the best model in (Zhang, Wei, and Zhao 2017) by +3.72% top-1 MIU accuracy. The +14.94 improvement over the base P2C conversion module demonstrates

that the online learning vocabulary is effective. The using of bi-direction LSTM encoder produces a notable boost of +2.41% accuracy. Our P2C model seizes a slight but significant improvement as when tuning the dimension of CWE, our model gives 71.32% top-1 MIU accuracy. The performance on TouchPal corpus is similar and even more obvious, our best setting achieves 14.35% improvements compared to the best baseline.

The P2C module of IME outputs a rank list, and then interface once displays five candidates by default. If users cannot find the target character in the top 5 candidates, they have to click the Page Down button to navigate answers, which involve additional keystroke expenses for users. Therefore, we list the top-5 accuracy contrast to all baselines with top-10 results, and the comparison indicates the noticeable advancement of our P2C model. To our surprise, the top-5 result on the People’s Daily approaches the top-10 accuracy of Google IME. On TouchPal corpus, our model with the best

setting achieves 89.7% accuracy, surpassing all the baselines.

## Analysis

### Effects of Online Updated Vocaburay

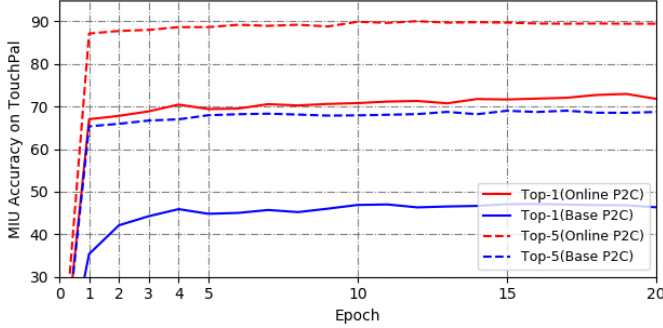


Figure 5: Training curves of top-1 and top-5 accuracies on TouchPal.

Figure 5 shows the changes of the MIU accuracy during the training process. For both top-1 and top-5 MIU accuracies, models with online vocabulary updating significantly outperform those without the updating throughout the entire training. Especially, online P2C gives top-1 MIU accuracy comparable to top-5 MIU accuracy given by the base P2C module, which suggests a great inputting efficiency improvement from introducing the online updating mechanism.

Figure 4 expounds the adaptivity of our online P2C, in which we feed a joint corpus that is interlately extracted from test corpora of the People’s Daily and Touchpal to the base P2C model and record the top-1 MIU accuracy per group after 2 epochs online vocabulary learning with batch size 1. We see that online P2C distinctly adapts the corpus change at the joint part. On the contrary, the base P2C which works offline performs stably only on its in-domain segments.

### Effects of Vocaburay Selection

As an adaptive vocabulary is used in our decoder, it may result in a very large vocabulary to encumber the decoder with efficiency. Therefore, in practice, we need to control the size of the vocabulary for acceptable decoding speed. However, pruning the vocabulary in any way will surely hurt the performance due to all items in the adaptive vocabulary added with a good reason. Figure 6 illustrates the relation among accuracy and decoding speed. The accuracies nearly does not get decreased with high enough decoding speed when taking 88.9% full vocabulary in our system.

### Effects of Word Filtering for CWE building

As we mentioned in Section , P2C conversion quality depends on the CWE mechanism which will benefit from an appropriate filtration ratio. As shown in Table 4, when the filter ratio equals to 0.9, the accuracy reaches the top. We notice two observations. First, pure word-level representation

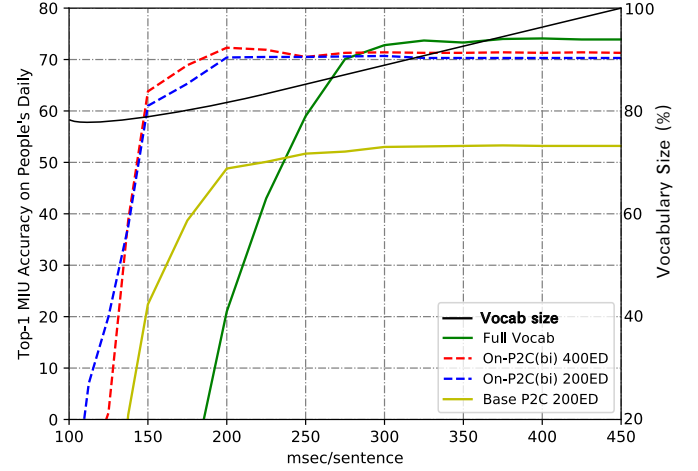


Figure 6: MIU accuracy versus decoding speed on CPU.

is more efficient for P2C tasks than character-level which only achieves 66.3% accuracy. Second, omitting partial low frequency word is instrumental in establishing word-level embedding. Actually, when building word embeddings, rare words function no more than noise. If the rare words are not initialized properly, they would also bias the whole word embeddings. Therefore, we more incline to make character-level embedding to represent rare word, and build CWE embeddings for others.

## User Experience

(Jia and Zhao 2013) proposed that the user-IME interaction contains three steps: pinyin input, candidate index choice and page turning. In Table 3, the 89.7% top-5 accuracy on TouchPal means that users have nearly 90% possibilities to straightly obtain the expected answer in the first page (usually 5 candidates per page for most IME interface setting), so that user experiment using IMEs can be directly measured by KySS. Table 5 shows the mean KySS of various models. The results indicate that our P2C conversion moduel further facilitates the interaction.

## Conclusion

This paper presents the first neural P2C converter for pinyin-based Chinese IME as to our best knowledge. We adopt an online working-style seq2seq model for the concerned task by formulizing it as a machine translation from pinyin sequence to Chinese character sequence. In addition, we incorporate an online vocabulary updating algorithm for further performance enhancement by tracking users behavior effectively. The evaluation on standard linguistic corpus and true inputting history show the proposed methods indeed greatly improve user experience in terms of diverse metrics compared to commercial IME and state-of-the-art traditional model.



## References

- [Cai et al. 2017] Cai, D.; Zhao, H.; Zhang, Z.; Xin, Y.; Wu, Y.; and Huang, F. 2017. Fast and accurate neural word segmentation for Chinese. In *ACL*, 608–615.
- [Chen and Lee 2000] Chen, Z., and Lee, K. F. 2000. A new statistical approach to Chinese pinyin input. In *ACL*, 241–247.
- [Chen, Wang, and Zhao 2015] Chen, S.; Wang, R.; and Zhao, H. 2015. Neural network language model for Chinese pinyin input method engine. In *PACLIC-29*, 455–461.
- [Chen, Wu, and He 2012] Chen, L.; Wu, X.; and He, J. 2012. Using collocations and K-means clustering to improve the N-pos model for Japanese ime. In *COLING*.
- [Chen 2003] Chen, S. F. 2003. Conditional and joint models for grapheme-to-phoneme conversion. In *Eurospeech 2003 - INTERSPEECH 2003*, 2033–2036.
- [Cho et al. 2014] Cho, K.; Merriënboer, B. V.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 1724–1734.
- [Choi, Cho, and Bengio 2017] Choi, H.; Cho, K.; and Bengio, Y. 2017. Context-dependent word representation for neural machine translation. *Computer Speech and Language* 149–160.
- [Emerson 2005] Emerson, T. 2005. The second international Chinese word segmentation bakeoff. In *SIGHAN*, 123–133.
- [Hatori and Suzuki 2011] Hatori, J., and Suzuki, H. 2011. Japanese pronunciation prediction as phrasal statistical machine translation. In *IJCNLP*, 993–1004.
- [Huang et al. 2018] Huang, Y.; Li, Z.; Zhang, Z.; and Zhao, H. 2018. Moon ime: Neural-based Chinese pinyin aided input method with customizable association. In *ACL: System Demonstrations*, 140–145.
- [Jean et al. 2015] Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2015. On using very large target vocabulary for neural machine translation. In *ACL*.
- [Jia and Zhao 2013] Jia, Z., and Zhao, H. 2013. Kyss 1.0: a framework for automatic evaluation of Chinese input method engines. In *IJCNLP*, 1195–1201.
- [Jia and Zhao 2014] Jia, Z., and Zhao, H. 2014. A joint graph model for pinyin-to-chinese conversion with typo correction. In *ACL*, 1512–1523.
- [Jiampojamarn, Cherry, and Kondrak 2008] Jiampojamarn, S.; Cherry, C.; and Kondrak, G. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*, 905–913.
- [Jiang et al. 2007] Jiang, W.; Guan, Y.; Wang, X. L.; and Liu, B. Q. 2007. Pinyin to character conversion model based on support vector machines. *Journal of Chinese Information Processing* 21(2):100–105.
- [L’Hostis, Grangier, and Auli 2016] L’Hostis, G.; Grangier, D.; and Auli, M. 2016. Vocabulary selection strategies for neural machine translation. *arXiv preprint arXiv:1610.00072*.
- [Lin and Zhang 2008] Lin, B., and Zhang, J. 2008. A novel statistical Chinese language model and its application in pinyin-to-character conversion. In *CIKM*, 1433–1434.
- [Luo and Roukos 2008] Luo, X., and Roukos, S. 2008. A self-adaptive lexicon construction algorithm for Chinese language modeling. *Acta Automatica Sinica* 34(1):40–47.
- [Luong and Manning 2016] Luong, M.-T., and Manning, C. D. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL*, 1054–1063.
- [Luong, Pham, and Manning 2015] Luong, M. T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, 1412–1421.
- [Mi, Wang, and Ittycheriah 2016] Mi, H.; Wang, Z.; and Ittycheriah, A. 2016. Vocabulary manipulation for neural machine translation. In *ACL*, 124–129.
- [Mikolov et al. 2013] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [Mori et al. 1998] Mori, S.; Tsuchiya, M.; Yamaji, O.; and Nagao, M. 1998. Kana-Kanji conversion by a stochastic model. *IPSJ SIG Notes* 2946–2953.
- [Mori, Takuma, and Kurata 2006] Mori, S.; Takuma, D.; and Kurata, G. 2006. Phoneme-to-text transcription system with an infinite vocabulary. In *COLING-ACL*, 729–736.
- [Neubig et al. 2013] Neubig, G.; Watanabe, T.; Mori, S.; and Kawahara, T. 2013. Machine translation without words through substring alignment. In *ACL*, 165–174.
- [Okuno and Mori 2012] Okuno, Y., and Mori, S. 2012. An ensemble model of word-based and character-based models for Japanese and Chinese input method. In *CoLING*, 15–28.
- [Verwimp et al. 2017] Verwimp, L.; Pelemans, J.; Hamme, H. V.; and Wambacq, P. 2017. Character-word LSTM language models. In *EACL*, 417–427.
- [Wu et al. 2018] Wu, Y.; Wu, W.; Yang, D.; Xu, C.; Li, Z.; and Zhou, M. 2018. Neural response generation with dynamic vocabularies. In *AAAI*.
- [Yang, Zhao, and Lu 2012] Yang, S.; Zhao, H.; and Lu, B.-l. 2012. A machine translation approach for Chinese whole-sentence pinyin-to-character conversion. In *Proceedings of the 26th Asian Pacific conference on language and information and computation*, 333–342.
- [Zhang et al. 2014] Zhang, J.; Utiyama, M.; Sumita, E.; and Zhao, H. 2014. Learning hierarchical translation spans. In *EMNLP*, 183–188.
- [Zhang, Wei, and Zhao 2017] Zhang, X.; Wei, C.; and Zhao, H. 2017. Tracing a loose wordhood for Chinese input method engine. *arXiv preprint arXiv:1712.04158*.
- [Zhao et al. 2006] Zhao, H.; Huang, C.-N.; Li, M.; and Kudo, T. 2006. An improved Chinese word segmentation system with conditional random field. In *SIGHAN*, 162–165.
- [Zhou et al. 2016] Zhou, J.; Cao, Y.; Wang, X.; Li, P.; and Xu, W. 2016. Deep recurrent models with fast-forward connections for neural machine translation. In *TACL*.