KySS 1.0: a Framework for Automatic Evaluation of Chinese Input Method Engines*

Zhongye Jia and Hai Zhao†
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
jia.zhongye@gmail.com, zhaohai@cs.sjtu.edu.cn

Abstract

Chinese Input Method Engine (IME) plays an important role in Chinese language processing. However, it has been subjected to lacking a proper evaluation metric for a long time. The natural metric for IME is user experience, which is a rather vague goal for research purpose. We propose a novel approach of quantifying user experience by using keystroke count and then correspondingly develop a framework of IME evaluation, which is fast and accurate. With the underlying linguistic background, the proposed evaluation framework can properly model the user behavior as Chinese is input through English keyboard. It is helpful to point out a way to improve the current Chinese IME performance. [1]

1 Introduction

Chinese IME is a software solution that enables user to input Chinese into computer with a reasonable size keyboard, by mapping Chinese characters into English letter combinations. Nowadays the majority of Chinese IMEs are pinyin based. Pinyin is originally designed as the phonetic symbol of a Chinese character, using Latin letters as its syllable notation. For example, the pinyin of the Chinese character "爱"(love) is "ài".[2] There are only less than 500 pinyin syllables in standard modern Chinese, compared with over 6,000 commonly used Chinese characters, which leads to serious ambiguity for pinyin-to-character mapping. Other IMEs using various mapping scheme more or less share this same ambiguity problem, although some of them such as five-stroke IME[3], may have lower ambiguity but they are very difficult to learn. So at present pinyin IMEs are the most popular, we focus on them in this paper. Modern IMEs are *sentence-based*(Chen and Lee, 2000) to reduce the ambiguity, which means that the IME generates a character sequence upon a sequence of English letter inputs, e.g. pinyin syllables. This is a non-typical sequence labeling task, as English alphabet input is the original sequence and Chinese characters are target labels. IMEs usually utilize beam search incorporated with language model initialized by (Chen and Lee, 2000), then later extended by (Liu and Wang, 2002), (Lee, 2003), and (Zhang, 2007). There are also various methods using conventional sequence labeling techniques such as: support vector machine (Jiang et al., 2007), maximum entropy model (Wang et al., 2006), conditional random field model (Li et al., 2009) and machine translation (Yang et al., 2012a), etc. IME also attracts attention from major software and internet corporations, Microsoft[4], Google[5] and many others developed their own IME products.

Note that "sentence" from an IME's viewpoint is not the linguistic "sentence". It is actually the *Max Input Unit* (MIU), the longest consecutive Chinese character sequence inside a sentence. For example, the sentence "第51届ACL年会即将召开" (The 51st annual meeting of ACL will open soon)

*This work was partially supported by the National Natural Science Foundation of China (Grant No.60903119, Grant No.61170114, and Grant No.61272248), and the National Basic Research Program of China (Grant No.2009CB320901 and Grant No.2013CB329401).

†Corresponding author

[1] KySS is the abbreviation for: **Key**Stroke **S**core

[2] Chinese is a language with tone, but for nearly all pinyin IMEs, tone marks are omitted since it is inconvenient to input.

[3] http://www.wangma.com.cn/

[4] http://bing.msn.cn/pinyin/

[5] http://www.google.com/intl/zh-CN/ime/pinyin/

International Joint Conference on Natural Language Processing, pages 1195–1201,
Nagoya, Japan, 14-18 October 2013.

has 3 MIUs: "第", "届" and "年会即将召开". For more precise expression, we will use "MIU" or "character sequence" throughout this paper as far as possible, instead of "sentence".

Chinese is used by the largest population in the world, and IME is necessary for all Chinese computer users. However according to our best knowledge there are no comprehensive and quantified metrics for evaluating its performance. The existing evaluation metrics for IME (Chen and Lee, 2000; Yang et al., 2012a) only focus on the character sequence generation, by adopting typical sequence labeling measurements such as character-wise precision and recall, sequence error rate and so on. However IME is an application deeply involving human-computer interaction, user behavior ought to be taken into account for IME evaluation(Zheng et al., 2011). Rather than merely judging the performance of character sequence generation, a good IME evaluation system should properly model the user behavior.

Compared with other typical evaluation systems for NLP tasks, such as the well known machine translation evaluation system BLEU (Papineni et al., 2002) and the summarization evaluation system ROUGE (Lin, 2004), they measure the closeness of machine translation/summarization and human results, using some n-gram co-occurrence statistics (Lin and Hovy, 2003), while our IME evaluation framework tries to quantify user experience during inputting Chinese. Existing keystroke based methods mostly focus on the keystroke duration and frequency pattern for biometric authenticate system security (Giot et al., 2009; El-Abed et al., 2012), instead of user experience modeling.

2 The Evaluation System

2.1 The User-IME Interaction

As introduced in Section 1, nearly all IMEs nowadays are "sentence based". However, IME does not always succeed to exactly generate the entire expected character sequence for an input letter sequence. Thus IMEs output a list of all possible *candidate* character sequences corresponding to the input sequence.

Suppose that the input pinyin sequence is $S_1S_2...S_n$ where S_i is a pinyin syllable with index i and n is number of pinyin syllables. The *best character sequence* has the same length as the input pinyin sequence, and is always in the first position of the list. Other output character sequences in the candidate list is given according to $S_1S_2...S_{n'}$ where n' is usually less than n. As n is usually large, it is unlikely that the best character sequence is completely correct, but for those shorter character sequences in the candidate list, user can always find that one of them may partially match his/her input target. If user has selected a candidate from the list to partially complete the input, for example, the character sequence for $S_1S_2...S_j$ has been determined by this selection, then IME will dynamically output a new list of character sequences for the rest pinyin sequence $S_{j+1}S_{j+2}...S_n$. User can continue to make the selection from the list until the desired input is accomplished. For a candidate, we define its position in candidate list as its *rank*, r. The best character sequence has $r = 0$. The candidate at the j-th position in the list has $r = j - 1$.

There are often dozens of candidates for a pinyin sequence, as the space of input interface is limited, candidates have to be shown in multiple pages. A *page* is part of the candidate list. User can make the choice from the page by pressing candidate ID $1, 2, \ldots, m$ in the current page. If the expected word/character does not occur in the current page, user has to press a "next-page"[6] key to see more candidates.

Suppose that one wants to input the MIU "年会即将召开"(The annual meeting will open soon) by typing the pinyin sequence "nian hui ji jiang zhao kai", a typical IME prompt window is like Figure 1a.

As shown in Figure 1a, the best character sequence is not completely correct for the input purpose. User has to pick up the 2nd candidate "年会"(annual meeting). Then the IME will update the window as Figure 1b. User can select the 3rd candidate "即将"(soon). The IME will update for the rest as shown in Figure 1c. The first candidate exactly completes the expected character sequence.

In the real world, user behavior is very difficult to predict. In order to alleviate the difficulty of user behavior modeling, an abstract assumption for user-IME interaction is proposed: all user input actions are only restricted to pinyin sequence input, candidate ID selection and page turning. In this manner, user inputs a sequence of pinyin then make a choice from the candidate list in the current page given by the IME, if the desired character se-

[6]Usually this "next-page" key is not the PageDown key, but mapped to some more convenient keys such as "+" or ".".

(a) The first IME window (b) The second IME window (c) The third IME window

Figure 1: IME windows for inputting an MIU

quence is not presented in the current page, then user has to press "next-page" and goes on until the target is met.

Under this proposed user behavior model, user input is divided in to two parts: 1. a sequence of alphabet keys for pinyin and 2. a selection action sequence of "next-page" and candidate ID keys. The first part of inputting alphabet keys for different IMEs will be always the same, thus it can be ignored for evaluation metric. The second part would be the essential difference among different IMEs. We define the sequence of ranks of candidates as *Rank Sequence*, For an ideal rank sequence , it is always just $\{0\}$. For our previous example in this section, the actual rank sequence is $\{1, 2, 0\}$.

2.2 Evaluating IME

To make use of the rank sequence, i.e. keystroke count, as the metric to evaluate an IME, we define a few terms as follows:

- $\mathcal{L}$: It is the length of MIU in characters.
- $\mathcal{P}$: It is the length of rank sequence. It measures how many parts the MIU is split into to accomplish the input. In the previous example, $\mathcal{P} = 3$ for "年会", "即将" and "召开". In the ideal situation, $\mathcal{P}$ for any MIU is 1 since the exactly expected character sequence is rank at the top.
- $\mathcal{R}$: It is the sum of all the elements in the rank sequence, i.e. the sum of ranks of candidates for each part. In our previous example, $\mathcal{R} = 1 + 2 + 0 = 3$. For the ideal situation, $\mathcal{R}$ is always 0.
- $\mathcal{R}_W$: It is the total keystroke without alphabet keys, by using the weighted sum of rank sequence:

$$\mathcal{R}_W = \sum_{i=1}^{k-1} \omega(r_i) + 1. \quad (1)$$

The weight function $\omega(\cdot)$ reflects the cost of pressing "next-page" and candidate ID keys. In the rest of this paper we will assume there are 5 candidates on each page which is the default setting for most existing IMEs. We also assume the keystroke cost of pressing numeric keys for candidate ID is 1 and the keystroke cost of pressing "next-page" is also 1, then the weight function is

$$\omega(r) = \lfloor \frac{r}{5} \rfloor + 1.$$

For example the 1st candidate on 3rd page has $\mathcal{R}_W = 1 \times 2 + 1 = 3$. $\mathcal{R}_W$ measures how many keys the user has to press i.e. how much effort the user has to make to accomplish inputting an MIU.

- KySS: It is the final evaluation score of an IME. Consider an evaluation corpus $\mathbb{C}$ with MIUs $\{m_1, m_2, \ldots, m_c\}$, for a certain IME, the i-th MIU m_i has $\mathcal{R}_W(i)$, then the KySS score for the actual IME on the corpus is defined as the ratio between the total $\mathcal{R}_W$ for the ideal IME and the total $\mathcal{R}_W$ given by the actual IME:

$$\text{KySS} = \frac{\sum_{m_i \in \mathbb{C}} \mathcal{R}_W^{ideal}(i)}{\sum_{m_i \in \mathbb{C}} \mathcal{R}_W(i)} \quad (2)$$

$$= \frac{\|\mathbb{C}\|}{\sum_{m_i \in \mathbb{C}} \mathcal{R}_W(i)}. \quad (3)$$

For an ideal IME, we have KySS $= 1$. For actual IMEs, $0 <$ KySS < 1. An IME with higher KySS is supposed to performan better.

3 Analysis on IMEs

3.1 Corpus

To build a corpus for evaluation, we extract 100,000 sentences from *China Daily* corpus[7]. We annotate it with pinyin sequence using the method in (Yang et al., 2012b). The corpus contains over 4 million characters, 87.6% of which are Chinese

[7]http://www.icl.pku.edu.cn/icl_res/default_en.asp

characters, other 12.4% are foreign letters, digits and punctuation. At last, a corpus with over 420,000 MIUs is built.

The IME for evaluation is sunpinyin[8] which is the state-of-the-art open-source Chinese pinyin IME on Linux developed by the former *Sun Microsystems, Inc.*.

The distribution of $\mathcal{L}$ for the evaluation corpus is shown in Figure 2. It can be seen that the length of

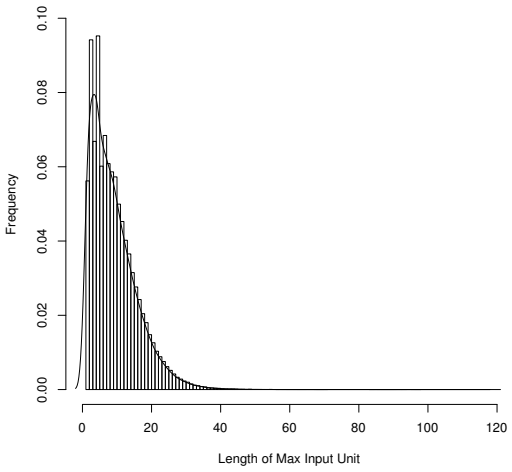


Figure 2: Distribution of $\mathcal{L}$

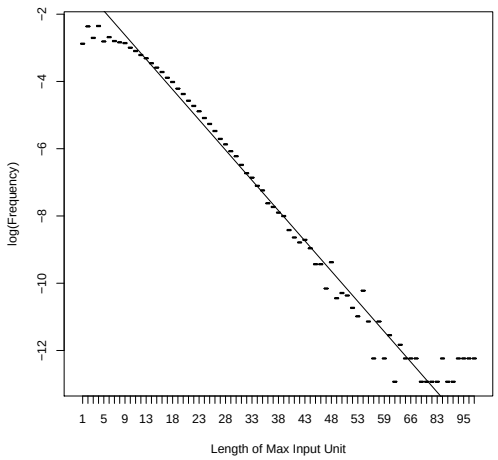


Figure 3: Linear fit on log-frequency over $\mathcal{L}$

MIUs roughly follows an exponential function distribution in statistics. A linear regression on log-frequency is made and it fits the actual data well. The result is shown in Figure 3.

[8]http://code.google.com/p/sunpinyin/

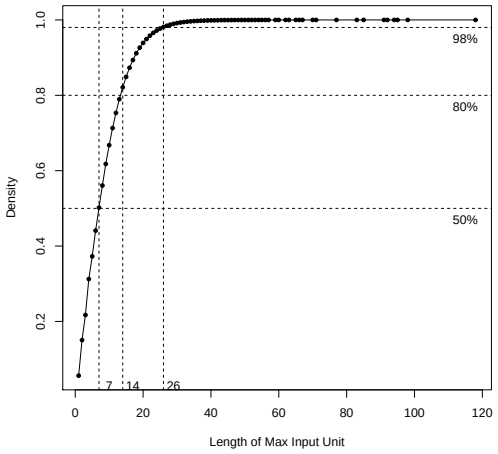


Figure 4: Cumulative distribution of $\mathcal{L}$

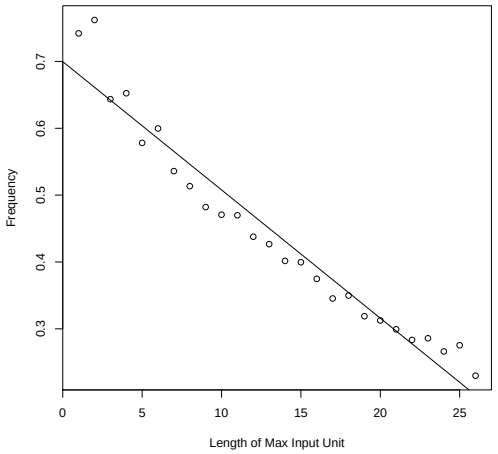


Figure 5: Rate of successful MIU generation

Although there are very long MIUs in the corpus, it can be observed from the cumulative distribution of $\mathcal{L}$ in Figure 4 that most MIUs are short ones.

Paying so much attention to the length statistics of MIUs is necessary, because the core task of IME, character sequence generation, is heavily affected by the sequence length. The rate of successful character sequence generation decays linearly with the increment of $\mathcal{L}$ as shown in Figure 5.

It also can be seen from Figure 4 that nearly 50% of MIUs are between 7 and 26 characters long, but as shown in Figure 5 they suffer from the rate of successful character sequence generation less than 0.5, which will be our major concern later.

3.2 Position Matters: How Character Sequence Generation Fails

From our empirical observation over those cases in which IME fails to generate the completely correct MIU, we found something interesting. The histograms of $\mathcal{P}$ and $\mathcal{R}$ of MIUs with $\mathcal{L} = 10$ are shown in Figure 6, note that only unsuccessful results are plotted in the figure.

The interesting point is that a peak appears at $\mathcal{L}/2$ in the histograms, as is shown in dotted lines in Figure 6. All the cases that we observed with length ranging from 7 to 26 demonstrate this "*error peak*" property.

We extract MIUs, IME generated best character sequences and rank sequences of those cases. The underlying reason is discovered after a manual inspection: in those best character sequences, the last error is at the rear of MIU. In such cases user has to select candidate words one by one until the end of the MIU.

For example, while inputting the MIU "大理航站是个年轻的航站"(Dali airport is a new airport), the best character sequence generated by IME is "大力航展是个年轻的航展"(STRONG air show is a new air show). Its $\mathcal{L} = 11$. The selected candidates are: "大理"(Dali), "航站"(airport), "是个"(is), "年轻"(new), "的"(*meaningless function word*), and "航站"; so $\mathcal{P} = 6$. And the rank sequence is $\{2, 1, 1, 1, 1, 1\}$ so $\mathcal{R} = 7$. So when errors occur at the rear of MIU, we have $\mathcal{P} \approx \mathcal{N}_W$, where $\mathcal{N}_W$ is the number of words in MIU. For most of the real circumstances, those correctly generated words in the front part of MIU achieve a very high rank r, often the highest among candidates, i.e., $r = 1$ so $\mathcal{R} \approx \mathcal{P} \approx \mathcal{N}_W$. In sunpinyin, candidates except for the best character sequence are words queried from a internal dictionary. An important linguistic issue is that average length of Chinese words is about 1.8 characters, nearly two character long (Zhao et al., 2006). At last we obtain $\mathcal{N}_W \approx \mathcal{L}/2$, which explains the "error peak" at the middle position.

3.3 Improving IMEs with Evaluation Results

We propose a method to make use of the "error peak" for better IME performance. Among all those MIUs with the "error peak" problem, we can see an annoying situation is that all words except for the last few ones are successfully generated. In order to correct one or two words, user has to select word by word all the way through the character sequence. To avoid unnecessary selection, the IME can cut the front part of the generated character sequence and make it an independent candidate so that user can accomplish inputting as many words in one selection as possible. We suggest five cutting policies as the following:

- **Weakest**: Cut at the position where the conditional probability is smallest.
- **Mean, LtoR**: Cut at the position where the probability is lower than the mean of conditional probabilities of the sequence, scanning from left to right.
- **Mean, RtoL**: Similar as previous but scanning from right to left.
- **Halfway**: Cut at exactly at the middle position.
- **Fixed**: Cut a fixed length sequence. We experiment on length from 1 to 10 and found the best length is 3.

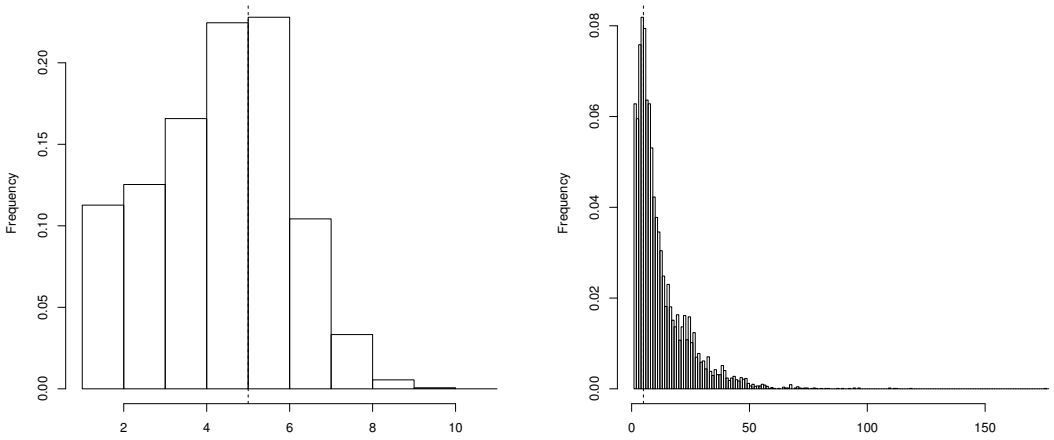
The KySS for each policy is shown in Table 1. The baseline is the original algorithm used by sunpinyin without any cutting. To our surprise, the dummy **Fixed-3** and **Halfway** cutting performs best with a more than 10% boost over the baseline due to its robustness. The **Fixed-3** and **Halfway** policy can always stably cut out a reasonable length of MIU. The problem with **Mean, RtoL** is that it is likely to cut at the end of MIU thus the head that it cuts out may still include errors. On the contrast, the **Mean, LtoR** policy tends to cut out too few words. And the **Weakest** policy is easy to fail because it often cuts a sequence with a low conditional probability but actually being correct.

Policy	KySS
Baseline	27.67%
Weakest	29.41%
Mean, LtoR	29.21%
Mean, RtoL	29.06%
Halfway	30.71%
Fixed-3	31.30%

Table 1: KySS of each policy

4 Conclusion and Future Work

In this paper, a novel evaluation framework for Chinese IME, KySS, is proposed by effectively modeling user behavior during Chinese input. This evaluation framework aims to fast and accurately evaluate various IMEs from the view of user experience. It uses keystroke count as core metric.



(a) Histogram of $\mathcal{P}$

(b) Histogram of $\mathcal{R}$

Figure 6: Histogram of $\mathcal{P}$ and $\mathcal{R}$, with $\mathcal{L} = 10$

With the help of the framework we preliminarily propose a sequence cutting strategy to enhance the current IME.

The real world IME and user behavior can be very complicated. In this paper, we make a simplified assumption that all user input is correct. Unfortunately it may contains typos. And the IME may have prediction feature, i.e. generating character sequence longer than the input pinyin sequence. We may include those in our future work.

References

Zheng Chen and Kai-Fu Lee. 2000. A New Statistical Approach To Chinese Pinyin Input. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 241–247, Hong Kong, October.

Mohamad El-Abed, Patrick Lacharme, and Christophe Rosenberger. 2012. Security evabio: An analysis tool for the security evaluation of biometric authentication systems. In *Biometrics (ICB), 2012 5th IAPR International Conference on*, pages 460–465. IEEE.

Romain Giot, Mohamad El-Abed, and Christophe Rosenberger. 2009. Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–6. IEEE.

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

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

L. Li, X. Wang, X.L. Wang, and Y.B. Yu. 2009. A conditional random fields approach to chinese pinyin-to-character conversion. *Journal of Communication and Computer*, 6(4):25–31.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

B.Q. Liu and X.L. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

X. Wang, L. Li, L. Yao, and W. Anwar. 2006. A maximum entropy approach to chinese pin yin-to-character conversion. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 4, pages 2956–2959. IEEE.

S. Yang, H. Zhao, and B. Lu. 2012a. A machine translation approach for chinese whole-sentence pinyin-to-character conversion. In *Pacific Asia Conference on Language Information and Computation*, pages 367–376, Bali, Indonesia, November.

S. Yang, H. Zhao, X. Wang, and B. Lu. 2012b. Spell checking for chinese. In *International Conference on Language Resources and Evaluation*, pages 730–736, Istanbul, Turkey, May.

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

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of PACLIC*, volume 20, pages 87–94.

Yabin Zheng, Lixing Xie, Zhiyuan Liu, Maosong Sun, Yang Zhang, and Liyun Ru. 2011. Why Press Backspace? Understanding User Input Behaviors in Chinese Pinyin Input Method. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Techologies*, pages 485–490, Portland, Oregon, USA, June. Association for Computational Linguistics.