Robustness Analysis of Adaptive Chinese Input Methods

Mike Tian-Jian Jiang[§⌘] **Cheng-Wei Lee**[⌘] **Chad Liu**[⌘] **Yung-Chun Chang**[❖⌘] **Wen-Lian Hsu**[⌘]
[§]Department of Computer Science, National Tsing Hua University
[❖]Department of Information Management, National Taiwan University
[⌘]Institute of Information Science, Academia
{tmjiang, aska, chadliu23, changyc, hsu}@iis.sinica.edu.tw

Abstract

This work proposes a novel metric, *Maximally Amortized Cost* (*MAC*), for cost evaluations of error correction of predictive Chinese input methods (IMs). With a series of real-time simulation, user correction behaviors are analyzed by estimating *generalized backward compatibility* of adaptive Chinese IMs. Comparisons between three IMs by using *MAC* with different context lengths report empirical factors of context length for improving predictive IMs. The *error-tolerance level*—*Futile Effort*, *Beneficial Effort* and *Utility*—of adaptive IMs is also proposed and analyzed.

1 Introduction

Most ideograph-based Asian languages consist of thousands of characters, making it impractical to create keyboards along the same style as alphabetic languages. In response, most modern systems come with built-in tools called input methods (IMs) for transforming multiple keystrokes into single ideographs. IMs are often categorized into "radical-based" or "phonetic-based" methods. With radical-based IMs, users construct characters by typing the composing radicals or strokes. Alternatively, phonetic-based IMs rely on phonetic transcriptions of ideographs, where users create characters by typing in the approximate spellings of their syllables. In the case of homographs or homophones, users are given a choice, and the proper character is selected and entered.

Besides desktop environments in Asian languages, IMs are also essential in any language for ambiguous keyboards that have more than one character or letter assigned to each key, resulting in some uncertainty about the intended symbol when a key is pressed. Ambiguous keyboards gain attentions because of mobile computing, which has limited space. Also, such keyboards expand the communication possibilities for users with physical disabilities who have insufficient motor facility to operate a full-size keyboard. Two methods enable ambiguous keyboards to access a large set of characters, and these differ depending on who performs the disambiguation. First, there is the multi-tap method or non-predictive method, in which the user disambiguates using multiple keystrokes to uniquely indicate a character. In the case of a full-size keyboard, additional keystrokes such as those applied through the CapsLock key are a kind of multi-tap entry. The second approach uses a predictive method, in which the system disambiguates and presents a list of ordered candidates from which the user chooses. For example, predictive IMs on the 12-key ITU-T keypad of mobile phones such as T9 and LetterWise have been studied with human-computer interaction (HCI) metrics that measure text entry performance in terms of speed and accuracy, in order to quantitatively analyze user experiences of different IMs (MacKenzie *et al.*, 2001; Silfverberg *et al.*, 2000). All of these studies, however, focus on alphabetic languages, and mostly English; thus far, HCI research on IM in other languages has been underdeveloped.

While various types of IM can be used with a keyboard, this work specifically examines the context of predictive phonetic-based methods for Chinese. Predictive phonetic-based IM not only facilitates word prediction and word or phrase completion, but also disambiguates homophones of syllables into characters. To date, most natural language processing (NLP) research on Chinese IMs has focused on these predictive phonetic-based approaches. Many researchers have applied n-gram language modeling (LM) and hidden Markov models to IMs, such as Chen *et al.* (2000), Gao *et al.* (2002), Wang *et al*. (2004), and Wu *et al*. (2003); Maximum entropy (Li *et al*., 2007) and conditional random fields (Xiao *et al*., 2009) have also been employed. While the studies above have made important contributions, they also assume fixed rules or stationary

Proceedings of the Workshop on Advances in Text Input Methods (WTIM 2011), pages 53–61,
Chiang Mai, Thailand, November 13, 2011.

probabilities. Developers of IMs, however, are expected to pay more attention to the increasing needs on personalization[1] and new word supplements via search engine logs[2] or social networks[3]. Only a handful of research papers to our knowledge explore adaptive language modeling of IM for Asian languages. Tanaka-Ishii *et al*. (2003) have examined corpus for vocabulary acquisition for Japanese in terms of reused words and unused words; Suzuki and Gao (2005) have proposed an e*rror ratio* corresponding to the number of newly introduced errors per each improvement after new training text was supplied. These two studies reflect a common expectation of IM users—*backward compatibility*—which means a word prediction that was previously correct should remain correct with new words recognized simultaneously.

This work intends to expand the approach towards *backward compatibility* using novel evaluation methods for Chinese predictive phonetic-based IM, by comparing text entry performance before and after user corrections of predictive IM-generated errors. Once an error is left uncorrected, it becomes noise to an IM with the ability to adapt. In addition, user corrections could be more complicated in predictive Chinese IMs than Japanese ones. When the user modifies some character, its surrounding characters often change automatically, because unlike Japanese, Chinese syntax does not have clear cues and orders of subject-verb-object typology. Thus, predictive Chinese IMs must rearrange the whole entered text to construct more likely context according to certain user modifications. After this kind of continuous automatic adjustment, user feedback is often too vague to interpret into exact word boundaries for adaptation, in terms of vocabulary acquisitions. It is considerably closer to daily usage of IM and more difficult than most previous works of adaptive IMs that acquire new information from correct and manually segmented transcriptions. This work suggests that a robust predictive Chinese IM should tolerate noisy user feedback during adaptation, in addition to the *backward compatibility* mentioned earlier.

To improve understanding of these situations, this work reviews existing performance evaluation metrics related to IMs, and then proposes extensions of these metrics for predictive and adaptive Chinese IMs, especially in cases of *generalized backward compatibility* and *error-tolerance level* for cost and influence. This work also develops a platform that is fully capable of simulating user-IM interaction, so as to collect data for quantitative comparison of various uses or different IMs. The proposed evaluation metrics and simulation results provide helps for further NLP investigation of predictive phonetic-based IM on error-tolerant adaptation and conduct pilot tests to report empirical factors before engaging in labor-intensive corpus annotations and human-participated HCI research.

2 Properties of Chinese Predictive IM with Adaptation Ability

2.1 Online Implicit User Feedback

Recent Chinese predictive IM products provide several ways for users to leave feedback on vocabulary acquisition. These methods practice in two different perspectives: online vs. offline and explicit vs. implicit. Online feedback indicates that an IM collects unknown words or re-ranks known words based on the user's current actions, while offline feedback means an IM extracts similar information via user-provided content or logs. When the user indicates their preferences directly, an IM receives explicit feedback; otherwise it must interpret user-IM interactions for implicit feedback. While offline and explicit feedback can be modeled as reinforcement learning or through the research of Tanaka-Ishii *et al*. (2003) or Suzuki and Gao (2005), our goal is to explore the relatively unfamiliar territory of implicitly online user feedback.

2.1.1 IM Adaptation Procedure

First, extending the definition from Tanaka-Ishii *et al*., (2003) any predictive IM with adaption abilities lets the user enter text continuously in five stages:

1. The user enters an ambiguous source keystroke string.
2. The IM retrieves candidate chunks corresponding to the source string from its built-in database and the user's profile.
3. The IM sorts these candidate chunks and composes most likely chunks to a target string, according to a particular evaluation function.

[1] Google Pinyin's privacy terms (in Chinese), http://www.google.com/intl/zh-CN/ime/pinyin/privacy.html
[2] Sougo Cell dictionary, http://pinyin.sogou.com/dict/
[3] Social IME, http://www.social-ime.com/

4. The user modifies the target string by choosing candidate chunks in case the IM's prediction is not entirely correct.
5. The IM adapts the user's modifications with context as implicit online feedback for user profiling.

One may argue that user's modifications can be accumulated as logs for lazy evaluation as offline feedback. According to some Chinese IM product's customer service reports (personal communication), however, users expect their modifications to be adapted as soon as possible to avoid repeat modifications for the same error cases. This expectation motivates this work to investigate real-time solutions of online feedback.

2.1.2 User Adaptation Habit

One intuitive and ideal solution of online feedback involves applying early evaluations of Move-to-Front (Bentley *et al*., 1986) and Prediction by Partial Match (Bell *et al*., 1990) techniques on modified chunks with context. In our experience, however, users may also adapt to an IM's performance and develop habits to correct just one chunk and then submit the target string immediately, which leaves fewer contexts for an IM to analyze. To overcome this situation, some IMs analyze unmodified target strings for more information, which can be misleading if the user has left some incorrect chunks. Eventually users will face a dilemma: typing more chunks to feed an IM for better adaptive predictions but encountering more errors. Hence this work studies properties of IM regarding the trade-off between cost and benefit of error correction.

2.2 Error Correction Evaluation Metrics

In order to understand the role of *Amortized Cost* that will be defined later in this section, it is first useful to examine previous research on error correction by describing well-known evaluation metrics for text entry and considering their shortcomings. To avoid confusion, all metrics use the notations formerly introduced by Soukoreff and MacKenzie (2003) as follows:

- Presented text (*P*) is text that participants were required to enter by the experiment, and |*P*| is the length of *P*;
- Transcribed text (*T*) is the final text entered by the participant, and |*T*| is the length of *T*;
- Input stream (*IS*) is the text that contains all keystrokes performed while entering *P* and |*IS*| is the length of *IS*;
- Correct (*C*): the number of correct characters in *T*;
- Incorrect Not Fixed (*INF*): the number of unnoticed errors in *T*;
- Incorrect Fixed (*IF*): keystrokes are those in *IS* that are not editing keys (*F*), and which do not appear in *T*;
- Fixes (*F*): are keystrokes in *IS*, which are edit functions, modifier keys, or navigation keys.

2.2.1 MSD

Evaluating the accuracy of text entry involves more than simply comparing strings. Consider the following example:

```
P: the quick brown fox
T: the quixck brwn fox
```

The notion of minimum string distance (MSD), which is the minimum number of primitives—insertions, deletions, or substitutions—needed to transfer one string to another, is introduced to deal with such a situation (Soukoreff *et al*., 2001). In this case, *P* and *T*'s MSD is 2. The idea of MSD error rate is to find the smallest number of operations to transform *T* to match *P*, and then to calculate the ratio of that number to the larger of |*P*| and |*T*|. The MSD error rate is defined as

$$MSD\,ErrorRate = \frac{MSD(P,T)}{\overline{S_A}} \times 100\%,$$

where $\overline{S_A}$ is the mean length of the alignment strings. MSD can only provide information about the remaining *T*, because errors corrected by the editing process can no longer be observed.

2.2.2 KSPC

In contrast to MSD, it is possible to observe corrected errors by logging all keystrokes as *IS*. From *IS*, a new metric, key-strokes per character (KSPC), is defined by MacKenzie (2001) simply as |*IS*| / |*T*|. KSPC sketches the effort required to correct errors without considering uncorrected errors. A large number of errors that only require low correction effort and a few errors requiring high correction effort may result in the same KSPC value. Although the keystrokes that send errors and keystrokes that correct errors are different, they are not differentiated by KSPC.

2.2.3 Unified Error Metrics

After observing the shortcomings of the MSD error rate and the KSPC value, Soukoreff *et al*. (2003) proposed a unified error metric that logs IS in the same way as KSPC and then classifies the keystrokes to analyze *T*. The MSD is only concerned with *INF*, while KSPC only reports the sum of *IF* and *F*. The *Total Error Rate* is a unified method, which recognizes all keystrokes of *INF* and *IF* and measures the ratio of the total number of incorrect and corrected characters as

$$TotalErrorRate = \frac{INF + IF}{C + INF + IF} \times 100\%.$$

The MSD error rate and KSPC statistic can be defined in terms of the keystroke taxonomy as

$$MSDErrorRate = \frac{INF}{C + INF} \times 100\%;$$

$$KSPC \approx \frac{C + INF + IF + F}{C + INF}.$$

For example, once a user corrected the error "`brwn`" of *T* to form "`the quixck brown fox`" as *T'*, *TotalErrorRate*(*T'*), *MSDErrorRate*(*T'*), and *KSPC*(*T'*) will be (2/18)%, (1/17)%, and 19/17, respectively.

2.3 Evaluation of Predictive IM

Predictive Chinese IMs consist of a display buffer for composition as a target string waiting for editing, and lists of candidate chunks for every potential editing position. These characteristics, which come from the complexity of languages that do not have delimiters (e.g. spaces) in their writing systems, such as Chinese and Japanese, are not captured by the metrics discussed above, because those metrics were originally designed for short text entry with alphabetic languages on handheld devices. It is therefore necessary to consider an alternative approach to overcome the shortcomings of existing metrics. In doing so, this work first examines long buffer variables and multiple candidate lists by reviewing Fitts' law and Hick's law before using them to create an improved evaluation metric.

2.3.1 Fitts' law

Fitts' law is a function of the distance to the final target and its size, and is used to predict the time required to move rapidly from a starting position to a final target area. Mathematically, Fitts' law can be formulated in several ways. One refined form, proposed by Soukoreff *et al*. (2003) is

$$t = a + b \log_2(d / w + 1),$$

where the average time *t* is taken to complete the movement, and *a* and *b* are empirical constants that can be determined by fitting a straight line to measured data. The distance *d* is from the starting point to the center of the target. The width *w* is of the target measured along the axis of motion. The term $\log_2(d/w + 1)$ represents the index of difficulty (*ID*) of the given task. Since a text entry task usually shifts the cursor by keystrokes rather than mouse movements, *ID* may link to the number of keystrokes directly.

2.3.2 Hick's law

When correcting typing errors, both the time taken by moving cursor and the time for candidate selection should be considered. Here, Hick's law,

$$t = a + b \log_2(n + 1),$$

describes the time, *t*, it takes users to make a decision as a function of the equal possible n choices they have, where *a* and *b* are empirical constants. The law hints some baseline points, but the realistic candidate selection time still needs to be measured via subject experiments. As far as we know, Hick's law has not been widely adapted to candidate selection for typing error correction of text entry tasks.

2.3.3 Maximally Amortized Cost

In previous work of Arif *et al*. (2009), text entry experiments are conducted with one of three error correction conditions, including *None*, *Recommended* and *Forced*. The participants are not allowed to correct any error in the *None* condition. On the other extreme, participants are forced to correct every error to keep *T* error free in the *Forced* condition. Lastly, participants are recommended to correct errors as they identify them in the *Recommended* condition. During the *None* condition, typists sometimes instinctively tried to correct their errors before they remembered that they could not. Such a failed error correction attempt takes a bit of time, as participants need to mentally recover and resume the original task. Again, during the *Recommended* condition participants tended to correct their errors almost the moment they made them (i.e. character level error correction), making this condition similar to the *Forced* condition.

In the end, Arif *et al*. did not find any relationship between the typists' entry speed and their instinctive attempt to correct errors. Therefore, the *None* and *Forced* conditions are not considered hereafter. Furthermore, this work argues that a more realistic condition of error correction

Situation	Fixed characters	*INF*	*IF*	*F*
S_0	none	INF_0	0	0
S_i	some	INF_i	IF_i	F_i
S_{all}	all	0	IF_{all}	F_{all}

Table 1. Three situations of errors correction

lies in the spectrum of motivations behind *Recommended* conditions. For the purpose of efficiency, a user may not correct most errors that occur during mobile phone texting or Internet chatting, but the same user is likely to try to make every word as effective as possible in situations of formal writing. When a predictive IM is involved, the user tends to find a compromise between efficiency and effectiveness according to the certain IM's performance, as mentioned in subsection 2.1.2 of users' habit, for example. In fact, technical news articles in China have even devised a conventional performance evaluation metric called "accuracy rate of the first suggested chunk[4] (首選詞正確率)". This has not been adopted in academic papers, since it lacks clear definitions for chunk and reference corpus. If the predictive IM adapts user behavior while the user adapts IM behavior simultaneously, feedback in-between could be very complicated. To model this phenomenon, situations are categorized, as shown in Table 1, and an information theoretic point of view is applied to define the *Amortized Cost* (*AC*) of text entry as follows:

$$AC = \frac{WatsedBandwidth}{UtilisedBandwidth} = \frac{\frac{INF + IF + F}{C + INF + IF + F}}{\frac{C}{C + INF + IF + F}} = \frac{INF + IF + F}{C}$$

where the basic measurement of the four categories is character. Some might argue that the metric *F* should be counted on keystrokes. However, each function/control keystroke *F* can successfully map to a virtual character unit as an information term. As long as the numerators and denominators are measured in the same unit, the definition is satisfied. Although Table 1 shows three situations, only situation S_0 is easy for automated simulation because it is unconcerned about methods of corrections.

In alphabetic text entry, if assuming the same amount of errors occurred and the user applied the same correction skill in different situations, one could design a keystroke logger to record all editing processes and find the boundary of *AC*:

$$NumOfIncorrectCharaters = INF_0 + IF_0$$
$$= INF_i + IF_i = INF_{all} + IF_{all}$$
$$\Rightarrow INF_0 = IF_{all}$$
$$AC_0 \le AC_1 \le AC_{all},$$
$$\Leftrightarrow \frac{INF_0}{C} \le AC_i = \frac{INF_i + IF_i + F_i}{C} \le \frac{IF_{all}}{C} + \frac{F_{all}}{C} = \frac{IF_0}{C} + \frac{F_{all}}{C}$$

Unfortunately, unlike alphabetic text entry, it is insufficient to map the metric *F* to as the same measurement of the keystroke or character one by one for Chinese IMs, as with other metrics. For example, a backspace keystroke can be used to either erase a Chinese character or a phonetic character, and thus runs into trouble when evaluating its cost. For this reason, this work defines the metrics average correction penalty (*p*), average correction reward (*r*) and then another *AC* of modification (AC_m) instead of the original term F_{all} / C as

$$p = \frac{t_H \times INF_0 + t_F \times \max(ID)}{C + INF_0}, r = \frac{C}{C + INF_0},$$
$$\Rightarrow AC_m = \frac{p}{r} = \frac{t_H \times INF_0 + t_F \times \max(ID)}{C},$$

where *ID* is calculated by the distance moving to the furthest wrong word needing correction; t_H describes the time for selecting candidates measured by Hick's law; t_F represents the time for moving a cursor through *ID* based on Fitt's law. From these variables, a *Maximally Amortized Cost* (*MAC*) is proposed as follows:

$$MAC = \frac{INF_0}{C} + AC_m = \frac{INF_0}{C} + \frac{t_H \times INF_0 + t_H \times \max(ID)}{C}.$$

Here, *MAC* can be viewed as a metric to estimate user experiences of Chinese predictive IMs using automated simulation, which will be demonstrated in the next section.

2.4 Generalized Backward Compatibility

Tanaka-Ishii *et al*. (2003) argue that the major drawback to predictive IM is related to dictionary use; a user cannot enter vocabulary not registered in the dictionary. They presume that missing vocabulary should exist within the user's text, depending on the user's context. In order to analyze how vocabulary is reused when a user edits text, they investigate how the reused word rate changed according to the offset of a text, by marking the text at the offset of 0.5 KB and counting the reused word rate in the 1 KB window. Their results suggest that context is provided by 70% to 80% of the vocabulary and the story evolves through the rest. From this observation, they suspect that typical users reuse 70% to 80% of their vocabulary only after an offset window of several KB. Based on this previous work, simulations where text is typed repeatedly should

[4] ZOL reports (in Chinese),
http://soft.zol.com.cn/103/1033537.html
http://soft.zol.com.cn/132/1320458.html

be representative enough for adaptation of Chinese predictive IMs.

Suzuki and Gao (2005) present comparative experiment results on four techniques of adaptive LM for IMs. Their evaluation of four techniques is unique in that they go beyond simply comparing those techniques in terms of character error rate (*CER*); they measure the distance between background and adaptation domains by using a metric of distributional similarity, and attempt to correlate it with the *CER* of each adaptation method. They also propose a novel metric for measuring the side effects of adapted models using the notion of *backward compatibility*. The *error ratio (ER)* is introduced for estimating side effects, which is defined as $|E_A| / |E_B|$, where $|E_A|$ is the number of errors found only in the newly adapted model, and $|E_B|$ the number of errors corrected by the new model. Intuitively, *ER* captures the cost of improvement of certain adaptation method, corresponding to the number of newly introduced errors per each improvement.

Arif *et al*. have observed that error correction involves both human-specific elements and system-specific elements; for example, the time to verify a correction, and the key sequence required for replacing a wrong character, respectively. On one hand, users usually immediately verify what they have typed and correct errors right away, i.e. character-level correction. On the other hand, users also chunk their input and verify the result only after typing a few characters or even the whole word as word-level correction. This observation is quite similar to common usages of predictive Chinese IMs. As determined by analysis of human error correction behavior, however, the predominant strategy for alphabet text entry is to use the backspace key for both character-level *and* word-level corrections. This situation is different from predictive Chinese IMs, in that users tend to move to particular positions and then correct Chinese chunks (i.e. *de facto* words from the user's perception) by substitution.

This work expands the concept of *backward compatibility* to indicate a considerably more general and continuous scenario: previous corrections must not only remain correct after adaptation, but also new manual corrections made during adaptation should come into effect as soon as possible and remain correct as long as possible. For *generalized backward compatibility* (*GBC*) of adaptive Chinese IMs, a diagram from Arif *et al*. (2010) is modified to introduce new factors that represent intentional user skip error correction as Figure 1.

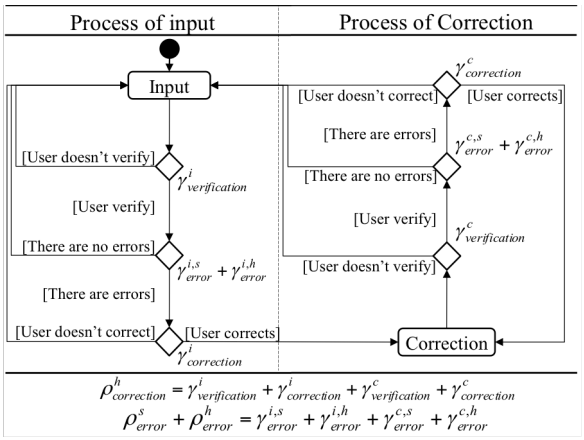

Figure 1. Activity diagram of user correction

Unlike Arif *et al*., who focused on how errors from non-predictive text entry systems (ρ^s_{error}) affect user experiences, this work is interested in how human correction behaviors ($\rho^h_{correction}$) influence accuracy of adaptively predictive IMs. The γ values stand for components of ρ at certain decision point. For instance, $\gamma^{i,h}_{error}$ represents the chance of human error occurred during the process of input. In order to test and demonstrate the ability of proposed evaluation methodology, this work conducts a simulation of three IM products.

3 Simulation

Three products of adaptively predictive IMs, named IM-A, IM-B, and IM-C, are used in the simulation. The presented text P consists of 4,000 sentences, containing 39,469 words retrieved from the Academia Sinica Balanced Corpus (ASBC) (Chen *et al*., 1996). Two independent variables are simulated: context length in terms of character and $\rho^h_{correction}$.

The context length k is for different strategies of word-level correction. Since there is not yet a consensus on the Chinese word-hood debate, the number of words is calculated by characters as context length k in this work. It is interesting to observe how IMs are influenced by these different strategies. The simulation is designed so that if $|T|$ is shorter than k, errors occurring in T will not change. Otherwise, the simulation will chop the first k characters of T to form a substring, denoted as T', and then process T' in the same way. For example, in a simulation with context length 3, "ab" remains intact, while "abcdefgh" is processed separately in three substrings "abc", "def", and finally "gh."

The factor $\rho^h_{correction}$ simulates human correction behavior. Here, errors are classified into two types: IM prediction error (ρ^s_{error}) and human typing error (ρ^h_{error}). The simulation simplifies

the sum of ρ^{s}_{error} and ρ^{h}_{error} as the ratio of corrected errors.

To simulate the actual typing process, the presented text *P* is converted into related keystrokes. Common transcription methods of Chinese characters are Bopomofo (also known as Zhuyin) and Pinyin. This simulation uses Bopomofo. There are many keyboard layouts for Bopomofo, such as Daqian (大千), Eten (倚天) or Hsu's (許氏). This work applies Daqian. Each Chinese character of *P* is transcribed into Bopomofo syllables and then transformed into Daqian keystrokes.

For *MAC*, estimating the time spent on candidate selection (t_H) and cursor movement (t_F) to the error needing correction is complicated. Many situations can occur during candidate selection, such as resorting to numeric keys to make a choice or seeking the correct word appearing on the next page of the candidate list, etc. Time also varies from person to person depending on how familiar they are with the IM. Clearly, it is impossible to quantify these two factors without having real-time user inquiries. This simulation assumes that the average time taken to choose a proper candidate is the same for every correction. Notably, the method of estimating t_F on a QWERTY keyboard is different from that of estimating t_F on a mobile keypad, because only thumbs are usually used in the latter case. In spite of this difference, it is observed that Chinese IM users rarely approach cursor movement with direct pointing devices such as a mouse. Thus, the value of t_F is simplified to the distance in terms of the character to which the cursor has to be moved.

The steps of the simulation consists of using different $\rho^{h}_{correction}$ to type all data of *P*, and then typing the same data again without correcting any error, so as to record and compare the character accuracy rate (*CAR*) after adaptation. For calculating *CAR*, *T* generated via particular IM is recorded and checked with *P*. For calculating *MAC*, the number of *C* and *INF* are counted while typing. During the simulation, the adaptive features of the IMs are enabled. Before the simulation, the adapted user profile of each IM is cleaned to ensure that the IM's *CAR* is unbiased.

3.1 Result

Figure 2 displays the comparison of *MAC* between the three IMs. For IM-A, IM-B, and IM-C between context lengths 1 and 4, their *MAC*s rapidly decline. IM-A's *MAC* continues to decrease slightly after context length 4 and the curve of the trend became relatively flat after context length 8; IM-B's *MAC* slowly increases during context 4 to context length 11 but there is an aberrant peaks at context length 12; IM-C's *MAC* generally draws a curve similar to IM-B's.

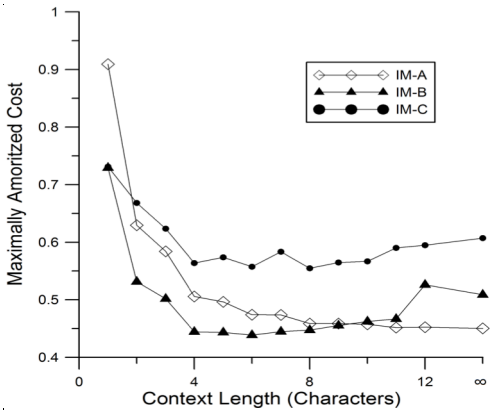

Figure 2. Comparison of *MAC*

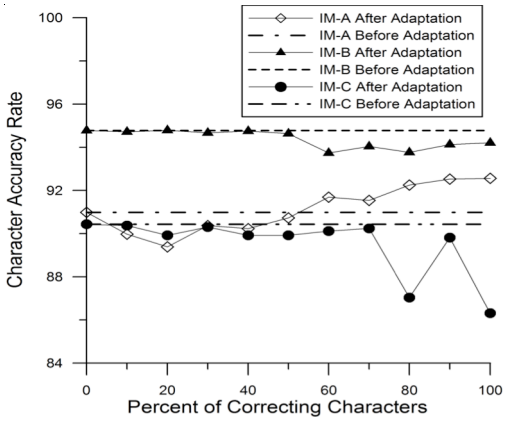

Figure 3. *GBC* at Context Length 6

Figure 2 shows $\rho^{h}_{correction}$ effects at context length 6 that is in the middle of relatively stable curves with low *MAC* for IM-A, IM-B, and IM-C, according to previous results. Instead of using *ER*, it is found sufficient to compare *CAR*s before and after adaptations in order to analyze the *GBC* of IM-A, IM-B, and IM-C. While the more corrections the user made the better adaptation IM-A performs, IM-B and IM-C show lower *GBC* when the user corrects more than 50% of errors.

4 Discussion

4.1 Empirical Factors of Context Length

According to Figure 2, the balanced choice of context length for IM-B and IM-C, in terms of *MAC* managing the trade-off between correction costs and context-provided benefits, is around 6 characters. This result suggests that it is possible to improve IM-B or IM-C by maintaining the size of a chunk for prediction and adaptation to 6 characters to save users' precious time in cursor

movement and candidate selection, without significantly decreasing accuracy. In the situation of IM-A, however, the ideal window size can expand to 8-13 characters, since the tail of its curve is smoother than IM-B and IM-C. Such difference is conjecturally related to their respective prediction and adaptation algorithms.

4.2 Error-Tolerance Level

The simulation result of Figure 3 show that at a context length of six characters, IM-A represents genuine *GBC* when the user corrects more than 50% of errors, but IM-B and IM-C encounter confusions when the user actively provides feedback. Since *GBC* involves user expectations of how fast a manually corrected chunk is adapted and how long it is sustained, this work provides a deeper analysis by defining three aspects of the *error-tolerance level* (*ETL*) as follows:

Futile Effort **(E^f)**: how many times a missing vocabulary in terms of chunk is typed by the user but still cannot be adapted by the IM;

Beneficial Effort **(E^b)**: how many times a missing vocabulary in terms of chunk is corrected by the user before it is adapted by the IM;

Utility **(U)**: how many times an adapted chunk is used before it is "forgotten" (because of the IM's limitation of memory space and/or adaptation algorithm, in general cases).

Table 2 and Table 3 show the corresponding maximums/averages of these three aspects for IM-A and IM-B, respectively, where chunks are sampled by character bi-grams and tri-grams automatically, so as to bypass the issue of Chinese word segmentation standards. The *ETL*s of IM-C are omitted in the interest of brevity and clarity, since IM-C's curves of *MAC* and *GBC* are similar to IM-B's.

$\rho^h_{correction}$	E^f_{max}	E^f_{avg}	E^b_{max}	E^b_{avg}	U_{max}	U_{avg}
10%	0	0.00	1	0.00	30	5.73
20%	2	2.00	1	1.00	22	8.30
30%	0	0.00	1	1.00	31	13.00
40%	4	2.40	3	1.45	51	12.05
50%	3	2.20	2	1.20	111	23.25
60%	2	2.00	6	2.50	57	20.85
70%	2	2.00	8	2.60	56	22.55
80%	5	2.35	9	3.00	35	18.75
90%	5	2.40	10	2.90	33	18.00
100%	5	2.25	18	3.55	29	16.50

Table 2. Error-tolerance level of IM-A

$\rho^h_{correction}$	E^f_{max}	E^f_{avg}	E^b_{max}	E^b_{avg}	U_{max}	U_{avg}
10%	0	0.00	1	1.00	33	8.00
20%	0	0.00	1	1.00	10	2.75
30%	2	0.00	2	1.05	33	9.95
40%	0	0.00	2	1.05	37	13.80
50%	2	2.00	2	1.20	31	10.45
60%	2	2.00	2	1.20	19	14.45
70%	3	2.13	4	1.70	28	11.65
80%	2	2.00	4	2.20	21	10.15
90%	5	2.45	3	2.25	24	12.10
100%	3	2.25	4	2.55	25	13.45

Table 3. Error-tolerance level of IM-B

For counts of manually corrected chunks that are never adapted as E^f, both IM-A and IM-B show that when the user puts more effort into correction, systems encounter more trouble with disambiguation. Statistics on reused counts of chunks U provide a different angle to *CAR* comparison of adaptation on *GBC*. IM-A holds adapted chunks better when the user has partially corrects input errors. Although IM-B seems to be relatively stable, it is unable to sustain its accuracy as long as IM-A. For quick responses and short-term memory of recently adapted chunks that are interpreted from E^b, IM-A and IM-B both get confused when the user corrects more frequently, and IM-A struggles harder than IM-B on the top-1 chunk. More specifically, for example, IM-A encounters frequent problems with Chinese homophones, where "his," "her" and "it" are all pronounced in the same disyllable, while IM-B seems to avoid any problems with this situation. Notably, IM-A's *CAR* series of *GBC* has correlation coefficients 0.49, 0.92, and 0.66 to E^f_{avg}, E^b_{avg}, and U_{avg}, respectively, while IM-B's has -0.78, -0.62, and -0.51.

5 Conclusions

This work proposes a novel metric for text entry evaluation of adaptively predictive Chinese IMs. The modification process of predictive Chinese IMs is quite different from that of alphabetic text entry (e.g. in English). Therefore, combining the time taken by cursor movements and candidate selections, and the *Amortized Cost* of information theory, the proposed metric, called the *Maximally Amortized Cost* (*MAC*), estimates the error correction cost of predictive Chinese IMs. A series of real-time simulation is then conducted, which approximates user correction behaviors for evaluation of *generalized backward compatibility* of adaptive Chinese IMs. Comparisons between three IMs using *MAC* with different context lengths report the appropriate context length as empirical factors for simulation and a possible direction to improve predictive Chinese IMs. This work has also suggested three aspects of *error-tolerance level—Futile Effort*, *Beneficial Effort*, and *Utility*—that could be useful for further investigation such as building reference corpus for shared tasks of IMs.

Acknowledgement

This research was supported in part by the National Science Council under grant NSC 100-2631-S-001-001, and the research center for Humanities and Social Sciences under grant IIS-50-23. Jaimie Lin, James Zhan, Jerry Lin, and Even Zheng contributed a lot of programming assistances. Dane Meyer is appreciated for his editorial assistance. The authors would like to thank anonymous reviewers for their constructive criticisms.

References

A. S. Arif and W. Stuerzlinger. 2009. Analysis of text entry performance metrics. Proceedings of the 2009 IEEE Toronto International Conference Science and Technology for Humanity, pp. 100-105.

A. S. Arif and W. Stuerzlinger. 2010. Predicting the cost of error correction in character-based text entry technologies. Proceedings of the 28th International Conference on Human Factors in Computing Systems, pp. 5-14.

T.C. Bell, J.G. Cleary, and I.H. Witten. 1990. Text Compression. Prentice Hall, New Jersey, USA.

J. L. Bentley, D. D. Sleator, R. E. Tarjan, and V. K. Wei. 1986. *A Locally Adaptive Data Compression Scheme*. Communications of the ACM, Vol. 29, No. 4, pp. 320-330.

K. J. Chen, C. R. Huang, L. P. Chang and H. L. Hsu. 1996. *Sinica Corpus: Design Methodology for Balanced Corpra*. Proceedings of the 11th Pacific Asia Conference on Language, Information, and Computation, pp.167-176.

Z. Chen, K. F. Lee, and M. T. Li. 2000. *Discriminative training on language model*. Proceedings of the Sixth International Conference on Spoken Language Processing.

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

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, Vol. 6, No.4, pp.25-31.

I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner. 2001. *LetterWise: prefix-based disambiguation for mobile text input*. Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology, pp. 111-120.

I. S. MacKenzie and K. Tanaka-Ishii. 2007. Text Entry Systems: Mobility, Accessibility, Universality. Morgan Kaufmann, San Fransisco, USA.

M. Silfverberg, I. S. MacKenzie, and P. Korhonen. 2000. *Predicting text entry speed on mobile phones*. Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 9-16.

R. W. Soukoreff and I. S. MacKenzie. 2003. *Input-based language modeling in the design of high performance input techniques*. Proceedings of Graphics Interface 2003, pp. 89-96.

R. W. Soukoreff and I. S. MacKenzie. 2003. *Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric*. Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 113-120.

R. W. Soukoreff and I. S. MacKenzie. 2001. *Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic*. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems, pp. 319-320.

H. Suzuki and J. Gao. 2005. *A comparative study on language model adaptation techniques using new evaluation metrics.* Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 265-272.

K. Tanaka-Ishii, D. Hayakawa, and M. Takeichi. 2003. *Acquiring vocabulary for predictive text entry through dynamic reuse of a small user corpus*. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pp. 407-414.

X. Wang, Q. Chen, and S. Daniel. 2004. *Mining Pinyin-to-character conversion rules from large-scale corpus: a rough set approach*. IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol. 34, Issue. 2, pp. 834 – 844.

G. Wu and Z. Fang. 2003 . *A method to build a super small but practically accurate language model for handheld devices*. Journal of Computer Science and Technology, Vol. 18, Issue. 6, pp. 747-755.

J. H. Xiao, B. Q. Liu, and X.L. Wang. 2007. *Exploiting Pinyin Constraints in Pinyin-to-Character Conversion Task: A Class-Based Maximum Entropy Markov Model Approach*. International Journal of Computational Linguistics and Chinese Language Processing, Vol. 12, No. 3, pp. 325-348.