Pinyin Aware: Adaptive Chinese Input Methods for Smartwatches

Reinhard Pointner
National Taiwan University
Taipei, Taiwan
reinhard.pointner@gmail.com

ABSTRACT

Smartwatches, wearable electronics, and other miniature devices are becoming increasingly popular. However, text input on such small devices remains a challenge due to small form factors, especially for non-Latin languages that require more complex text entry techniques. We implement and compare 3 fully functional soft keyboards for typing Mandarin Chinese using the Hanyu Pinyin system on the latest generation of circular smartwatch devices. We introduce 2 such novel adaptive keyboards, Growing Finals and Pinyin Syllables, which change dynamically based on the current input and the limited number of possible subsequent letters, optimize available screen size, and improve the user experience on small screens. Our evaluation is based on a user study with 15 participants and shows input speeds of around 19.4 CCPM (Chinese Characters per Minute) for Growing Finals and 18.5 CCPM for Pinyin Syllables after 20 minutes of practice. More than half of the participants ultimately preferred one of these input methods over the standard QWERTY keyboard due to reduced error rates and more efficient input mechanics.

CCS CONCEPTS

• **Human-centered computing** → **Text input**; *Touch screens*; Keyboards.

CHI'19, May 2019, Glasgow, UK
 The definitive Version of Record was published in *Proceedings of ACM CHI conference (CHI'19)*, https://doi.org/10.475/123_4.

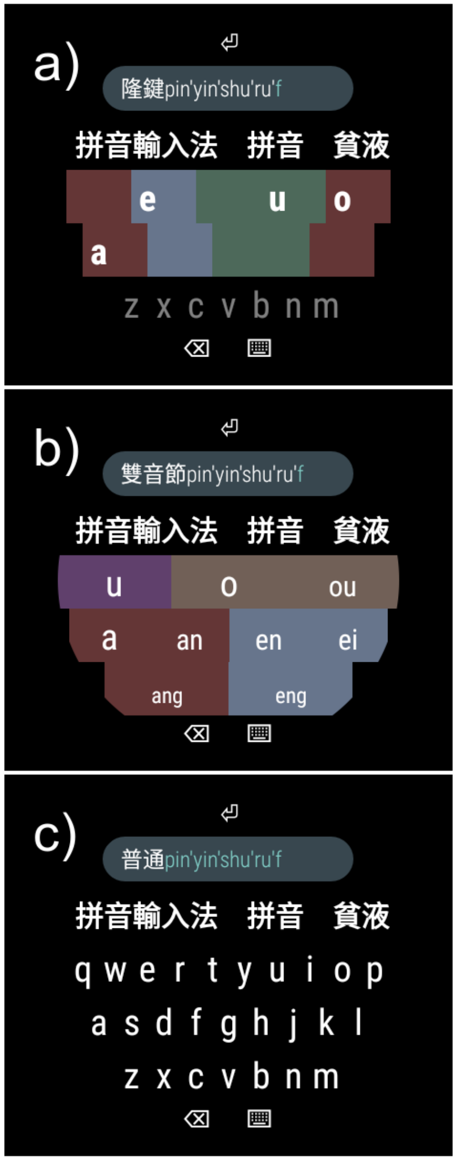

Figure 1: Keyboards for Chinese text input on smartwatches used in our study: (a) Growing Finals, (b) Pinyin Syllables and (c) Standard QWERTY.

KEYWORDS

Chinese text entry; pinyin; smartwatch; adaptive keyboard

ACM Reference Format:

Reinhard Pointner. 2019. Pinyin Aware: Adaptive Chinese Input Methods for Smartwatches. In *Proceedings of ACM CHI conference (CHI'19).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.475/123_4

INTRODUCTION

We design and implement 3 fully functional Chinese soft keyboards for modern circular smartwatches. We propose an adaptive quasi-QWERTY keyboard that enlarges the next possible keys to improve speed and accuracy, an adaptive 2-stage keyboard that allows any Pinyin syllable to be entered with a fixed number of 2 key strokes, and a standard QWERTY keyboard with 26 keys optimized for the small screen size which serves as a baseline. All 3 keyboard implementations use the same Pinyin-to-Chinese-Character conversion engine to allow for a fair comparison in our user study.

RELATED WORK

Text entry on small touch screen devices is a well explored topic [2][7][9][10], but input methods for more complex writing systems with more than 26 Latin characters [6] such as Chinese used by more than 1 billion native speakers are rarely studied.

Before the advent and widespread use of touchscreen phones, Liu, et al. have compared the performance of handwriting recognition (15 CCPM), a standard QWERTY keyboard (14 CCPM) and a static 2-stage consonant + vowel keyboard (9 CCPM) [4] for Chinese character input on a PDA with a stylus and explored Chinese text entry using a rotary input with RotaTxt (6 CCPM) [5]. The low input speed of these early works can partially be attributed to the lack of smart Pinyin-to-Chinese-Character conversion engines that could run on the limited hardware of the time.

More recently, Liu, et al. [3] improve upon the standard QWERTY keyboard for Chinese input on smartphones by adding a pie menu to each initial key to allow users to write a complete Pinyin syllable with one key press directly followed by multiple swipe motions (25 CCPM).

THREE PINYIN INPUT METHODS

We have designed and implemented 3 Chinese keyboards for circular smartwatches. Figure 1 shows our three prototypes. Each prototype has been implemented natively for Android Wear 2.0 and tested on an LG Watch Style. This smartwatch has a round 1.2" (30 mm) screen. Our implementation is open-source software, so that other researchers or developers can reuse and build upon our code.

Growing Finals. The Growing Finals keyboard (Figure 1a) exploits an intrinsic characteristic of the Pinyin romanization system: out the 26 letters of the Latin alphabet, 23 can appear as the first letter

Figure 2: Input sequence for the word 你好 (Pinyin: nǐ hǎo) using Growing Finals: 1) Press n 2) Press i 3) Press ' 4) Press h 5) Select 你好

in a Pinyin syllable, while the remaining 0 to 4 Latin letters of any Pinyin syllable are composed from only 9 unique Latin characters. Liu, et al. have previously exploited this characteristic in PinyinPie [3]. Further research revealed that after entering the first letter from a list of 23 options, entering the remaining letters of any Pinyin syllable requires no more than 6 unique possible next letters (including the syllable separator ' if necessary) in any input state. Hence, the Growing Finals keyboard will present a full QWERTY keyboard in its initial state and — depending on the input — adapt and enlarge possible next letters while preserving the general QWERTY layout (Figure 2). This allows for significantly larger keys after entering the initial letter for each syllable and thus improved input speed and accuracy according to Fitts' law.

Pinyin Syllables. Each Pinyin syllable can be seen as a combination of an initial sound and a final sound. Initials with similar sounds (e.g. b and p) tend to be combined with a similar set of final sounds (e.g. bang and pang) due to language characteristics. An analysis of the language model reveals that for each of the 26 initial sounds, there are between 2 and 24 (M = 15.4, SD = 5.6) possible final sounds (including no final sound for vowels such as a and e). The Pinyin Syllables input method implements a 2-stage adaptive keyboard layout based on Pinyin initials and finals which allows users to enter any Pinyin syllable with exactly 2 keystrokes. The first stage uses a pseudo-QWERTY keyboard layout to leverage existing familiarity with the standard QWERTY keyboard layout. The u, i, and v keys have been removed, and keys for sh, zh, and ch have been added so that each key on the initial keyboard corresponds to exactly one initial sound in the Pinyin language model. Once an initial key has been entered, the keyboard will adapt and display all possible finals for the current input state (Figure 3). Helping with visual search, finals with similar sounds are grouped by colour and placed near corresponding QWERTY keys if possible. Depending on the entered initial, the second stage may contain between 2 and 24 buttons allowing for significantly larger buttons in most cases with the added benefit that a single key press that completes the pinyin syllable may correspond to more than one Latin letter.

Standard QWERTY. We implement the standard QWERTY keyboard with support for Chinese character entry (Figure 1c) in addition to our two novel input methods to serve as a baseline for comparison. This keyboard design is used by almost all native speakers from China and language students from abroad that type Chinese characters on a computer or smartphone and thus very familiar to most user.

Common Features. Each keyboard has a text field that shows the text that has been entered so far. Partial Pinyin input that defines the current state of the keyboard is highlighted. Selecting Chinese characters from the list of suggested candidates will replace the current Pinyin input with the selected Chinese characters and reset the keyboard to its original state for the next input sequence. The

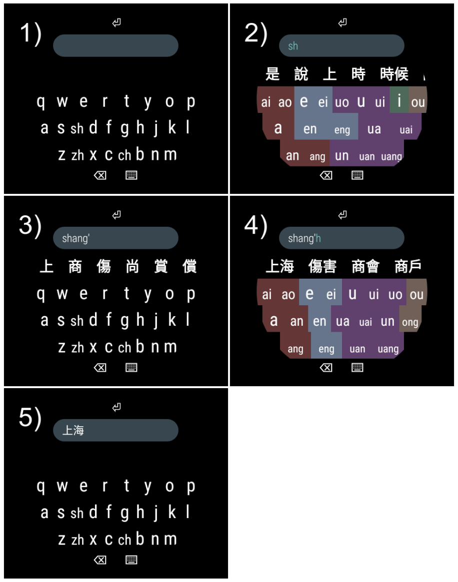

Figure 3: Input sequence for the word 上海 (pinyin: shàng hǎi) using Pinyin Syllables: 1) Press sh 2) Press ang 3) Press h 4) Select 上海

Figure 4: The setup of our user study.

rotating hardware button can be used to scroll through the list of Chinese character suggestions without occluding the screen. The DELETE key can be used to delete the most recently entered Latin letter or Chinese character. Cursor movements are not supported.

We use the cross-platform native library RIME [8] as high-performance Pinyin-to-Chinese-Character conversion engine via Java Native Interface (JNI) on our Android Wear prototype. RIME uses state-of-the-art language models and algorithms with support for intelligent character level, phrase level and full sentence level Chinese character prediction based on complete or partial Pinyin sequences.

EVALUATION

The aim of this study is to understand the characteristics and user preferences for each of our three keyboard prototypes. We conducted a user study consisting of text-copy tasks using sentences sampled from the NUS SMS corpus compiled by Chen, et al. [1] which consists of more than 30,000 Chinese short messages from Chinese users from all over China and is publicly available. For our study, we only consider short messages from users living in Beijing who are known to speak the most standard dialect of Mandarin Chinese in order to ensure that participants in our study are not confronted with unfamiliar local expressions or idioms. To focus only on Chinese sentences, we further restrict the study to samples that contain between 5 and 15 Chinese characters and no Latin letters or punctuation symbols.

For each of the three keyboards, participants had to enter 25 sentences. The total of $3 \times 25 = 75$ sentences for all keyboard prototypes was randomly sampled from a specific subset of the NUS SMS corpus [1] as explained above. These sentences contain only Chinese characters. All participants entered the same 75 sentences in the same order but the keyboard that was used for the first, second and third block of 25 sentences varied for each participant.

Results

During the trials, we recorded 1,125 samples of sentence input representing close to 30,000 individual input events and 8.3 hours of non-stop typing data. Table 1 shows the text entry performance metrics we derived from the typing data collected in our user study.

CONCLUSION

We compare three keyboard layouts for Chinese text entry on smartwatches. We implement a standard QWERTY keyboard and use it as a baseline to evaluate two novel adaptive keyboard layouts specifically designed for smartwatches that use unique features of the Chinese language model and Pinyin romanization system to allow for larger keys or keys that are mapped to multiple Latin letters reducing errors and number of keystrokes required to enter a given Chinese character.

Table 1: Mean Chinese Characters per Minute (CCPM), Total Error Rate (TER) and Keystrokes per Chinese Character (KSPCC) for each keyboard prototype.

	Growing Finals	Pinyin Syllables	QWERTY
CCPM	19.39 (4.3)	18.51 (4.0)	22.39 (5.9)
TER	7.70 (2.6)	9.49 (2.4)	12.21 (2.2)
KSPCC	3.72 (0.4)	2.98 (0.3)	3.79 (0.6)

REFERENCES

[1] Tao Chen and Min-Yen Kan. 2013. Creating a live, public short message service corpus: the NUS SMS corpus. *Language Resources and Evaluation* 47, 2 (01 Jun 2013), 299–335. https://doi.org/10.1007/s10579-012-9197-9
[2] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 615–620. https://doi.org/10.1145/2642918.2647354
[3] Ying Liu, Xiantao Chen, Lingzhi Wang, Hequan Zhang, and Shen LI. 2012. PinyinPie: A Pie Menu Augmented Soft Keyboard for Chinese Pinyin Input Methods. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 271–280. https://doi.org/10.1145/2371574.2371614
[4] Ying Liu, Kai Ding, and Ning Liu. 2009. Immediate User Performances with Touch Chinese Text Entry Solutions on Handheld Devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, Article 56, 2 pages. https://doi.org/10.1145/1613858.1613928
[5] Ying Liu and Kari-Jouko Räihä. 2008. RotaTxt: Chinese Pinyin Input with a Rotator. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. ACM, New York, NY, USA, 225–233. https://doi.org/10.1145/1409240.1409265
[6] Ken Lunde. 2008. *CJKV Information Processing* (2nd ed.). O'Reilly Media, Inc.
[7] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. https://doi.org/10.1145/2470654.2481387
[8] RIME. 2017. Rime Input Method Engine. http://rime.im/.
[9] Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-Chih Lin, Chen-Ting Wu, and Mike Chen. 2016. SwipeKey: A Swipe-based Keyboard Design for Smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 60–71. https://doi.org/10.1145/2935334.2935336
[10] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. COMPASS: Rotational Keyboard on Non-Touch Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 705–715. https://doi.org/10.1145/3025453.3025454