A Flick-based Japanese Tablet Keyboard using Direct Kanji Input

Yuya Nakamura[1] and Hiroshi Hosobe[2]
[1]*Graduate School of Computer and Information Sciences, Hosei University, Tokyo, Japan*
[2]*Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan*

Keywords: Text Entry Touch Screen, Touch Typing, Software Keyboard.

Abstract: Tablets, as well as smartphones and personal computers, are popular as Internet clients. A split keyboard is a software keyboard suitable for tablets with large screens. However, unlike other methods, the split keyboard has space in the center of the screen, which makes the part of the screen for displaying suggestions small and inconvenient. This paper proposes a Japanese input software keyboard that enables direct kanji input on a split flick keyboard. Once the user has mastered this keyboard, it allows the user to efficiently input Japanese text while holding a tablet with both hands. The paper presents an implementation of the keyboard on Android and reports the result of an experiment on its performance compared with existing methods. In addition, since direct kanji input generally takes time for users to learn, one of the authors by himself has conducted a long-term experiment to confirm the possibility of its mastery. For 12 months, both the input speed and the error rate have gradually improved.

1 INTRODUCTION

Japanese text is composed of Chinese-originated "kanji" characters and Japanese "kana" characters (Tamaoka, 2014). While a kanji character typically has a meaning, a kana character does not; instead, a kana character is associated with a speech sound. Computer users need to input both kanji and kana characters for Japanese text. Many Japanese keyboards use a conversion function from kana to kanji. One of the most common kana-kanji conversion methods used in Japan is the predictive conversion that uses past inputs as well as dictionaries. By contrast, direct kanji input lets users select kanji characters by themselves. It is believed that once a user mastered direct kanji input, the user can efficiently and effortlessly input Japanese text because of the unnecessity of judging conversion suggestions.

A split keyboard is a software keyboard for tablets that improves display space efficiency. It allows its user to put both hands at natural positions. The improvement of display space efficiency is good for multitasking. Split keyboards are often used with the QWERTY layout, and are adopted on Windows and iPad tablets (Knox, 2012; Apple Inc., 2019). Current split keyboards for Japanese input use common kana-kanji conversion methods such as predictive conversion. However, unlike other software keyboards, split keyboards have space in the center of the screen, which makes the part of the screen for displaying conversion suggestions small and inconvenient.

In Japan, there are about as many users of "flick" keyboards as QWERTY keyboard users. Flick is a gesture operation especially used for character input on touch-screen devices such as tablets and smartphones. Flick input usually makes its user to touch a key with a finger and then slide the finger upward, downward, to the left, or to the right for input.

This paper proposes a Japanese input software keyboard that enables direct kanji input on a split flick keyboard. We extend two-handed flick input (Nakamura and Hosobe, 2020) to allow direct kanji input. Once the user has mastered this keyboard, it allows the user to efficiently input Japanese text while holding a tablet with both hands. We propose two layouts: one uses elements of kanji characters called bushu; the other uses elements called on'yomi. Kanji characters are generally made up of dots and lines. The bushu-based layout uses a common collection of such dots and lines. The on'yomi-based layout arranges kanji characters by using their readings. They cover 2136 kanji characters called joyo-kanji, a standard set of kanji characters for social life in Japan.

For the objective evaluation of the proposed keyboard, we conducted comparative experiments. We compared the bushu-based layout, the on'yomi-based layout, two QWERTY software keyboards, and a flick keyboard. One of the QWERTY keyboards enabled

Nakamura, Y. and Hosobe, H.
A Flick-based Japanese Tablet Keyboard using Direct Kanji Input.
DOI: 10.5220/0010245800490059
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 2: HUCAPP*, pages 49-59
ISBN: 978-989-758-488-6

learning in predictive conversion for individual participants, and the other disabled it. We recruited 8 participants of ages ranging from 23 to 24. During the experiments, they were seated and held a tablet with both hands. The comparative experiments treated two kinds of input: sentence input and kanji conversion-required word input. The results showed that the on'yomi-based layout was faster in the input speed than the bushu-based layout. In addition, the results of subjective evaluation using the User Experience Questionnaire (UEQ) showed that the proposed keyboard was better in novelty than the existing methods although it was inferior in perspicuity, efficiency, and dependability.

Direct kanji input generally takes time for users to learn. Therefore, one of the authors by himself has conducted a long-term experiment to confirm the possibility of the mastery of the proposed keyboard using the bushu-based layout. The author has input five sentences every day. For 12 months, the input speed has increased, and the error rate has decreased.

2 RELATED WORK

In this paper, we extend Nakamura and Hosobe's bimanual flick software keyboard for tablets (Nakamura and Hosobe, 2020). It improved screen space efficiency by splitting a flick keyboard into the left and the right. However, it was not suitable for kana-kanji conversion because the conversion space generally extends from the left to the right edge of the screen. In this paper, we solve this problem by introducing direct kanji input.

Many Japanese input method uses predictive conversion, which presents conversion suggestions based on previously used words. For example, Ichimura et al. proposed a predictive kana-kanji conversion system (Ichimura et al., 2000). It used the current mainstream predictive conversion method that had been proposed several years before, and reduced users' keystrokes to 78 %.

Unlike predictive conversion that is used in many current Japanese input methods, direct kanji input lets the user select a kanji character. There are two types of direct kanji input: associative and non-associative. Associative direct kanji input has clear relationships between keystrokes and kanji characters. This method has the advantage of being more intuitive and easier to use (T-Code Project, 2003). On the other hand, non-associative direct kanji input does not have such clear relationships between keystrokes and kanji characters. From the user's point of view, it is a random key placement. For this reason, it is more difficult to use than the associative method. T-Code (T-Code Project, 2003) is one of the most famous methods of non-associative direct kanji input. T-Code uses a combination of two keystrokes on the QWERTY keyboard to enter a kanji character. There is no regularity in such key combinations, and the user needs to first learn them. The non-associative method takes longer time to learn than the associative method, but allows faster input (T-Code Project, 2003). This is because the associative method requires the user to associate kanji characters with keystrokes, but the non-associative method does not. However, whichever method is used, direct kanji input requires the user to more practice than kana-kanji conversion-based input. In this paper, direct kanji input keyboard was not included in comparative experiment. The reason is that we were not able to find any available software keyboards for tablets that used direct kanji input.

Research and development of input methods for Chinese characters are not limited to the Japanese language. Pinyin input is a widely used Chinese character input method that uses Chinese readings of characters (Li and Li, 2019). Cangjie is a direct Chinese character input method used in Hong Kong. In this method, users think of a Chinese character as a combination of parts. A keystroke corresponds to such a part, and a combination of keystrokes is used to input a Chinese character. Liu and Lin (Liu and Lin, 2008) proposed an extension of Cangjie to classify similar Chinese characters. Niu et al. (Niu et al., 2010) proposed Stroke++, a Chinese character input method for mobile phones, in which an input is made by combining bushu elements.

Various research has been done on keyboards for tablets. Sax et al. proposed an ergonomic QWERTY tablet keyboard (Sax et al., 2011). Bi et al. proposed a bimanual gesture keyboard to reduce display space and to shorten finger movement (Bi et al., 2012). Hasegawa et al. studied input of a software keyboard, with a focus on aging effects and differences between dominant and non-dominant hands (Hasegawa et al., 2012). Odell studied feedbacks of software keyboards (Odell, 2015). Takei and Hosobe proposed a Japanese kana input keyboard that input one character with two strokes by using 2×6 keys (Takei and Hosobe, 2018). Yajima and Hosobe proposed a Japanese software keyboard for tablets that reduced user fatigue (Yajima and Hosobe, 2018).

In Japan, much research on flick keyboards has been done. Sakurai and Masui proposed a QWERTY flick keyboard (Sakurai and Masui, 2013). This keyboard enabled input of Japanese kana characters and English letters without mode changes. Fukatsu et al. proposed an eyes-free Japanese kana input method called no-look flick (Fukatsu et al., 2013). This method enabled flick input for vowels and consonants in two keystrokes. Hakoda et al. proposed a kana input method using two fingers for touch-panel devices (Hakoda et al., 2013). This method was also an eyes-free Japanese input method, but enabled gesture input by two fingers.

Our software keyboard requires users to have more knowledge of kanji characters than existing methods, especially concerning their bushu and on'yomi elements. Therefore, the continuous use of our software keyboard could be regarded as a process of learning kanji or Chinese characters. Much research has also been done on learning Chinese characters. Dragon Tale (Plecher et al., 2018) is an adventure game in which players learn kanji characters. Kanev et al. (Kanev et al., 2012) proposed a 3D model to let students learn the elements of Chinese characters. In Japan, standard kanji education of children lasts for 9 years from the first year of elementary schools to the third year of junior high schools, which also has caused proposals of various methods of teaching kanji characters outside the research community.

3 JAPANESE CHARACTERS AND KEYBOARDS

3.1 Kana Characters

As previously described, Japanese text is composed of Chinese-originated kanji characters and Japanese kana characters. While a kanji character typically has a meaning, a kana character is associated with a speech sound. There are two kinds of kana characters called hiragana and katakana. Although they are used for different purposes, they correspond to each other; for each hiragana character, there is a corresponding katakana character, and vice versa. There are approximately 50 basic kana characters, which are further divided into 10 groups that are ordered, each of which typically consists of 5 characters. The first group is special because its 5 characters indicate 5 vowels that are pronounced "a," "i", "u", "e", and "o". The other 9 groups are associated with the basic consonants, "k", "s", "t", "n", "h", "m", "y", "r", and "w". A kana character in these 9 groups forms the sound that combines a consonant and a vowel. For example, the 5 characters of the "k" group are pronounced "ka", "ki", "ku", "ke", and "ko". This grouping of kana characters is basic knowledge of the Japanese language.

The "k," "s," "t," and "h" groups have variants called dakuon. Specifically, the dakuon variants of "k", "s", "t", and "h" are "g", "z", "d", and "b" respectively. In addition, the "h" group has another variant called handakuon, which is "p". Certain characters have variants that are written in smaller shapes. Sequences of kana characters can be expressed with the Roman alphabet by using the standard Japanese romanization system (ISO, 1989). This is widely used for computer users to enter Japanese text with alphabet keyboards such as QWERTY.

3.2 Elements of Kanji Characters

We explain bushu and on'yomi elements of kanji characters that we use in our software keyboard.

3.2.1 Bushu

Bushu, also called a radical, indicates an element of kanji characters. Kanji characters are generally made up of dots and lines. A bushu element is a common collection of such dots and lines. For example, Figure 1 shows kanji characters for a pine and cherry blossoms. The red boxes in the figure indicate their bushu elements. This type of bushu is called "kihen" and is typically used in kanji characters related to trees. Other kanji characters that use kihen correspond to, for example, a small forest and a bridge. The total number of bushu elements used in Japan is 214.

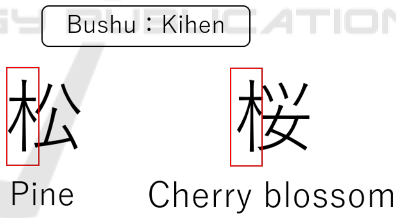

Figure 1: Examples of bushu elements.

3.2.2 On'yomi

Kanji characters typically have two kinds of readings, on'yomi and kun'yomi. The on'yomi of a kanji character indicates its old Chinese reading, while the kun'yomi indicates its Japanese reading. In general, the on'yomi of a kanji character does not make sense in Japanese while the kun'yomi does. For example, the kun'yomi of the kanji character for cherry blossoms in the Figure 1 is "sakura", which means cherry blossoms by itself. By contrast, its on'yomi, which is "ou", has no meaning in Japanese.

3.3 Japanese Keyboards

Figure 2 shows a typical Japanese flick keyboard. The main key layout is composed of 4 × 3 keys. If a user flicks a key to the left, upward, to the right, or downward with a thumb, the keyboard inputs a character corresponding to the direction (Figure 3). The enter key and the delete key are located on the right side of the keyboard. When either the “123” or the “ABC” key is pressed on the left side of the keyboard, the Japanese keyboard is replaced with the English letter keyboard or the number letter keyboard. A conversion space is located at the top of the keyboard. When a user touches a word, kana characters are converted to kanji or other characters. If a user touches the upward arrow, it will show other kanji candidates. In addition, most of Japanese input keyboards provide a function for learning conversions. This function allows the user to quickly select recently used characters. Predictive conversion is a function for predicting kana-kanji conversions from partial inputs. This function allows the user to more efficiently input text if the prediction is successful. However, it is often difficult to perform the predictive conversion from only a few kana characters.

Figure 2: Typical Japanese flick keyboard (which is available on iOS).

Figure 3: A typical input flow for a Japanese flick keyboard (which is available on iOS).

4 PROPOSED METHOD

4.1 Bimanual Flick

We propose a software keyboard for tablets that splits a flick keyboard into the left and right sides. It is based on the bimanual split flick keyboard proposed by Nakamura and Hosobe (Nakamura and Hosobe, 2020) that uses normal kana-kanji conversion. Instead of using such normal kana-kanji conversion, our new software keyboard introduces direct kanji input. Since the user can use the keyboard while holding the tablet with both hands, the advantages of split keyboards are not lost. If the user wants to input a kana character, the user only needs to flick a key on one side as with other splits keyboards.

A primary reason for developing a new direct kanji input method is that previous methods were designed for desktop computers. Although these previous methods could be adapted to newer devices such as tablets and smartphones, there has not been much research, and their effectiveness is unclear. Also, the proposed method might be applicable to the Chinese language. This is because Cangjie is similar to our bushu-based method and Pinyin input is similar to our on’yomi-based method (although Japanese and Chinese use different sets of standard Chinese characters, which would require extra efforts.)

A typical input flow for a kanji character is shown in Figure 4. This example inputs the kanji character meaning “cherry blossoms” previously shown in Figure 1. Figure 4-a shows a state in which no input is made. This kanji character uses the bushu element called “kihen”, which belongs to the “ki” group. The “ki” group further belongs to the “k” group. Therefore, the user first touches the “ka” key that represents the “k” group (Figure 4-b). Then the user flicks the finger to the left to show the “ki” group (Figure 4-c). The user looks for the kanji character for cherry blossoms, finding that it belongs to the top left key. Therefore, the user touches the top left key (Figure 4-d). Then the user flicks the finger to the right (Figure 4-e), which completes the input of the kanji character.

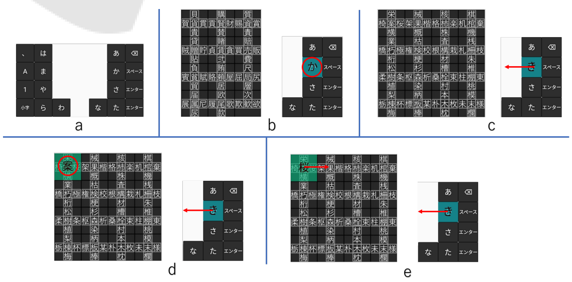

Figure 4: A typical input flow for a kanji character.

The number of kanji characters that can be input with this method is 2136. These kanji characters, called joyo-kanji, are indicated as the standard for using kanji characters in social life in Japan. According to a survey conducted by the Japanese Agency for Cultural Affairs, more than 96 % of the total number of kanji characters regularly used in Japanese society are joyo-kanji characters (Takeda, 2019). It was possible to cover more kanji characters in the proposed method. However, we decided that no more kanji characters were needed. Other kanji and kana characters can be input using the conversion function of the previous bimanual flick keyboard. It also should be noted that our method covers more characters than T-Code (T-Code Project, 2003), which covers 1600 characters.

4.2 Kanji Layouts

We propose two types of kanji layouts: the bushu-based layout and the on'yomi-based layout. In the comparative experiments, we compare these two methods with existing ones and evaluate their performance.

4.2.1 Bushu-based Layout

The bushu-based layout uses bushu elements of kanji characters. The layout is shown in Figures 5 and 6. Each cross in Figure 5 indicates a key that can be flicked upward, downward, to the left, and to the right, and the positions of the crosses correspond to the shape of the keyboard. The bushu elements that appear more than once in the figure are those that have many kanji characters. On the contrary, the bushu elements shown in Figure 6 have a few kanji characters. They are grouped together in the parts labeled with the numbers in Figure 5. In the case of kanji characters with the same bushu element, they are arranged by on'yomi from the top left. Also, since there is a limit on the size of the keyboard, four bushu elements with large numbers of characters are divided into two keys. In this case, they are located symmetrically.

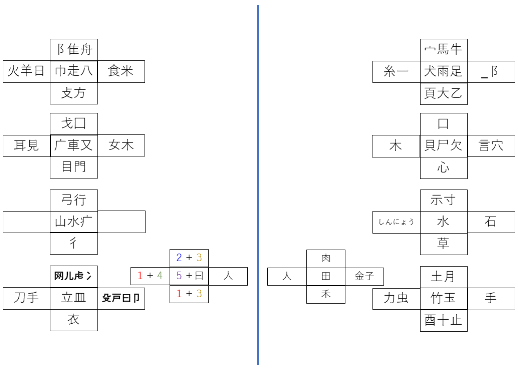

Figure 5: Placement of bushu elements.

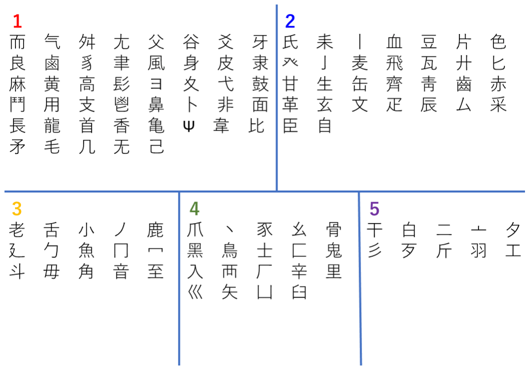

Figure 6: Bushu elements with a few kanji characters.

4.2.2 On'yomi-based Layout

The on'yomi-based layout arranges kanji characters by using their readings. Since dakuon is also present in on'yomi, the "k," "s," "t," and "h" groups can be replaced with the dakuon keyboard. Figure 7 shows the process of inputting the kanji characters of the "ka" group. If the user presses the top left key on the kanji keyboard, it is replaced with the dakuon keyboard. If the user presses the top left key on the dakuon keyboard, it is replaced with the original keyboard.

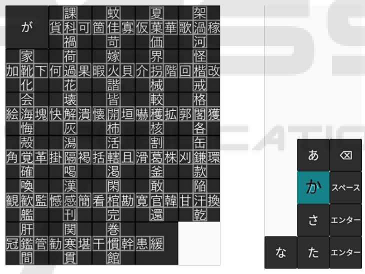

Figure 7: The on'yomi-based layout (for inputting a kanji characters in the "ka" group).

4.3 Direct Kanji Input

Our keyboard is based on direct kanji input. Previous direct kanji input methods were categorized into associative input and non-associative input. We regard the bushu-based layout as being semi-associative because it is neither associative nor non-associative in the original senses. It basically arranges kanji characters by bushu elements, but there is no perfect correspondence between kana characters and bushu elements. By contrast, the on'yomi layout is associative since there is correspondence between kanji characters and kana characters.

4.4 Key Arrangement

Our software keyboard is of size of 240-px height and 180-px width before conversion. This layout reduces the display space by 73 % in the portrait mode and by 83 % in the landscape mode, compared with the QWERTY keyboard with the maximum display width. The size and the position of the bushu-based layout are based on the heat map used in the design of the Windows 8 touch keyboard (Knox, 2012). This limited the bushu-based layout to 4×4. By contrast, the on'yomi layout does not have this limit. Therefore, in the case of the on'yomi-based layout, there may be much more kanji characters for a kana character than in the case of the bushu-based layout. In the case of the bushu-based layout, the maximum number of kanji characters for a kana character is 80. Since there are 50 kana characters in total, the total number of kanji characters that can be placed is 4000. We adopted 2136 joyo-kanji characters. Kana keys with a few kanji characters are composed of a 2×2 kanji layout. The keyboard in the bushu-based layout is of the maximum size of 320-px height and 320-px width on one side. The keyboard in the on'yomi-based layout is of the maximum size of 480-px height and 400-px width.

5 IMPLEMENTATION

We implemented the proposed software keyboard on an ASUS ZenPad 10 tablet (Android OS 7.0, 1920×1200-px screen) as shown in Figure 8. The keys are of 60×60 px, and the keyboard is placed symmetrically at the lower ends of the screen. In the proposed keyboard, its position was adjustable with a bar at the bottom of the screen. The red part in the figure indicates the conversion space that is used to enter non-joyo-kanji and katakana characters. The user can display different characters by swiping the conversion space to the left or to the right. The radio button in the center of the figure allows the user to change the key layout.

6 COMPARATIVE EXPERIMENTS

To evaluate the proposed software keyboard, we conducted an experiment on its comparison with existing software keyboards. We compared the bushu-based layout, the on'yomi-based layout, two QWERTY keyboards, and a flick keyboard. There are two types of QWERTY keyboards, one with the learning of predictive conversion enabled and one with the

Figure 8: Implementation of the proposed software keyboard.

learning disabled. The reason why the direct input method was not compared is that we were not able to find any available software keyboards for tablets that used direct kanji input. The learning of the predictive conversion is reset for each participant. The comparative experiments treated the landscape mode of each keyboard layout. In the proposed keyboard, its position was adjustable with a bar at the bottom of the screen. The position of the keyboard was set by each participant.

We recruited 8 participants who were Japanese university students and workers. Their ages ranged from 23 to 24, and all the participants were male. They were seated on a chair and held a tablet in the landscape mode with both hands. If participants were not able to reach the center of the keyboard in using the QWERTY, they were allowed to release their hands. The comparative experiments were composed of two input experiments and subjective evaluation. After the input experiments, we investigated subjective evaluation for each method. In addition to the UEQ, free descriptions were also collected.

We measured the input speed and the error rate. The input speed is measured by the number of characters per minute CPM, which is calculated as follows:

$$\mathrm{CPM} = \frac{\mathrm{T}-\mathrm{E}}{\mathrm{S}} \times 60 \qquad (1)$$

where T is the length of the input string, S is the input time, and E is the number of characters that were wrong. This means the number of characters typed correctly per minute. On the other hand, the error rate ER is calculated as follows:

$$\mathrm{ER} = \frac{\mathrm{IF}}{\mathrm{C}+\mathrm{IF}+\mathrm{INF}} \qquad (2)$$

where C is the total number of correct words, IF is the number of incorrect but fixed (backspaced) words, and INF is the number of incorrect (but not fixed) words. These equations are based on Bi et al.'s research (Bi et al., 2012), and we adjust them to Japanese character input.

The comparative experiments treated two kinds of input: sentence input and kanji conversion-required word input. In the following, we describe the details and the results of the two experiments.

6.1 Sentence Input

In the sentence input experiment, five input sentences were selected for each method from a book (Waragai, 2008) about learning joyo-kanji with example sentences. One of the QWERTY keyboards and the flick keyboard enabled learning in predictive conversion, but the sentences were specific to each method, and therefore the learning was limited to the inside of a method. Before the experiment, participants warmed up with a few sentences for each method. They started the experiment by pushing the start button on the upper left corner of the screen, and moved to the next sentence by pushing the enter key. A target sentence was displayed on the text field at the top of the screen. The sentences included in the list were of about 20-character length and mixed kana and kanji characters.

Since the bushu-based layout generally takes long time for users to learn, and also since the short warm-up before the experiment was not sufficient for the participants' learning, a support function was provided. It consisted of two hints: a hint for the bushu element of a kanji character and a hint for the position of a kanji character. The hint for bushu was displayed by pressing the hint button in the upper right corner of the screen (Figure 8). When the button was pressed, the bushu elements of kanji characters in the target sentence were displayed one by one. The hint for a kanji position was shown in Figures 5 and 6. The participants were able to look at this diagram if they did not know the positions of kanji characters.

6.2 Result of the Sentence Input Experiment

The results of the input speeds and the error rates are shown in Figures 9 and 10 respectively. In the charts, "QWERTY_ON" and "Flick" indicate the QWERTY keyboard and the flick keyboard with learning in predictive conversion respectively. A higher input speed is better, and a high error rate is worse. The results show large differences between the proposed methods and the existing methods (shown as the three bars on the right sides of the charts). The ANOVA on the two proposed methods also showed a significant difference in the input speeds ($p < 0.05$), but not in the error rates. Among the existing methods, the flick keyboard was the fastest, and the two QWERTY keyboards were of about the same speeds. Also, the flick keyboard was the highest in the error rates, and the two QWERTY keyboards showed about the same error rates.

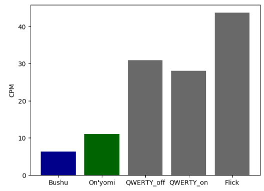

Figure 9: Input speeds.

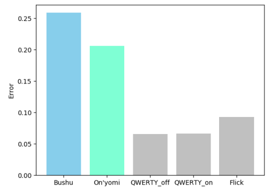

Figure 10: Error rates.

6.3 Kanji Conversion-required Word Input

The experiment on the kanji conversion-required word input treated the same layouts as the sentence input experiment. Its procedure was almost the same as that of the sentence input experiment, and the only difference was what the participants input. In this experiment, the participants input words written in kana characters at the top of the screen and converted them into kanji characters. The experiment was conducted after the sentence input experiment and there was no warm-up time. Five words were used for each method, and when a participant pressed the enter key, the next word was displayed. The target word was the word that appeared in the sentence input experiment. For this reason, the hints used in the sentence input experiment were not used in the word input.

6.4 Result of the Kanji Conversion-required Word Input Experiment

The results of the input speeds and the error rates are shown in Figures 11 and 12 respectively. The results show large differences between the proposed methods and the existing methods. The ANOVA on the two proposed methods showed a significant difference in the input speeds ($p < 0.05$), but not in the error rates. Among the existing methods, the QWERTY keyboard with learning in predictive conversion and the flick keyboard were the fastest and almost comparable. The three existing methods showed similar low error rates.

6.5 Subjective Evaluation

The subjective evaluation was performed by using the UEQ (Schrepp et al., 2017) to investigate the two proposed methods and the QWERTY keyboard with learning. Figures 13 and 14 showed the results of

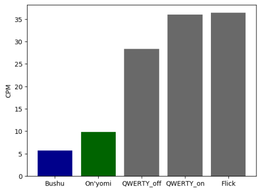

Figure 11: Input speeds.

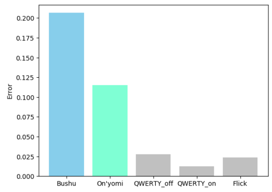

Figure 12: Error rates.

the subjective evaluation. In the bushu-based layout, the UEQ showed an excellent rating for novelty although the ratings of the other items were low. In particular, ratings for perspicuity and efficiency were very low. The on'yomi-based layout was rated higher than the bushu-based layout except novelty. When the on'yomi-based layout and the existing methods were compared, the existing methods were better in perspicuity, efficiency, and dependability, and the on'yomi-based layout was better in novelty.

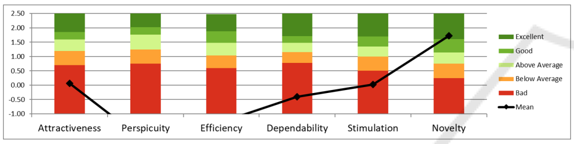

Figure 13: Result of the UEQ on the bushu-based layout.

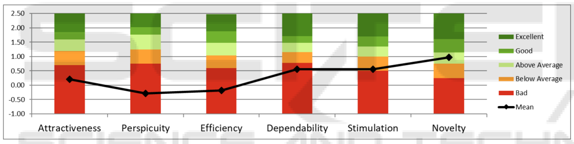

Figure 14: Result of the UEQ on the on'yomi-based layout.

The comparison of the on'yomi-based layout and the existing methods is shown in Figure 15. Both the QWERTY and the flick keyboard obtained higher ratings in perspicuity, efficiency, and dependability. The proposed method was rated higher in novelty.

7 LONG-TERM EXPERIMENT

7.1 Method

Direct kanji input is a method that takes time to learn. Therefore, one of the authors by himself conducted an experiment to confirm its mastery by using the proposed method for 12 months. The experiment consisted of the input of sentences. Every day the author entered five sentences chosen at random from 602 sentences in the literature (Waragai, 2008). The 602 sentences contain 68 % of the joyo-kanji characters. The author is a 23-year-old male graduate student in the field of computer science. In the experiment, the author was seated on a chair and held a tablet with

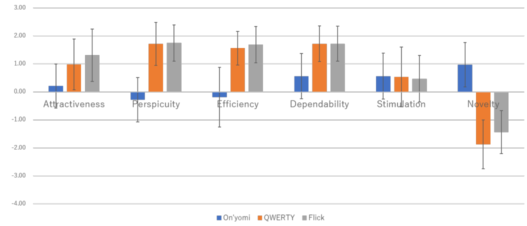

Figure 15: Result of the comparison of the on'yomi-based layout, the QWERTY keyboard, and the flick keyboard in the UEQ.

both hands. The experiment was conducted with the bushu-based layout, and a search function was used when the location of a kanji character was not known.

7.2 Result

Figures 16 and 17 show the results of the input speeds and the error rates respectively. In both charts, the main line indicates the average value for a day. The green bars represent the maximum and the minimum values for the day. For comparison, the author entered the same sentences by using the QWERTY keyboard on the ASUS ZenPad 10 (red line) and Nakamura and Hosobe's bimanual flick keyboard (Nakamura and Hosobe, 2020) (purple line). The QWERTY keyboard on the ASUS Zenpad 10 also enabled learning in predictive conversion. According to the charts, both the input speeds and the error rates gradually improved day by day. The reason why the bimanual flick keyboard had a high error rate was because it used the backspace key to make adjustments during the conversion.

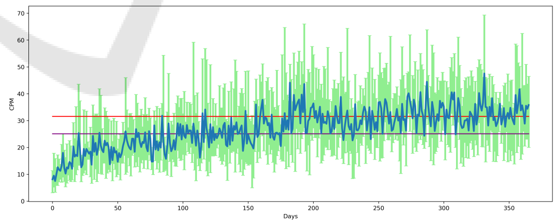

Figure 16: Input speeds (12 months).

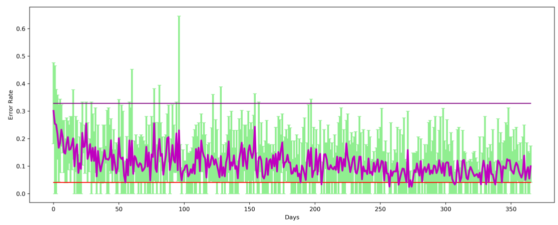

Figure 17: Error rates (12 months).

8 DISCUSSION

8.1 Comparative Experiment

8.1.1 Sentence Input

The result of the sentence input experiment showed that the proposed methods were inferior to existing methods in terms of the input speeds. There was also a significant difference in the input speeds between the proposed methods. We think that this is because the bushu-based layout is a semi-associative direct kanji input method and the on'yomi-based layout is associative. The semi-associative method takes longer time for users to learn than the associative method, but can be expected to enable faster input. The experiment tried to compensate for the difference by providing hints, but it was not successful.

The result also showed that the proposed methods were inferior to the existing methods in terms of the error rates. However, there was no significant difference between the proposed methods in terms of the error rates. We think that the error rates of the on'yomi-based layout in the sentence input experiment were higher than its error rates in the kanji conversion-required word input experiment because the given sentences were written in kanji and kana characters. In this case, if a participant could not read kanji, he would have trouble in input.

Therefore, to compare input speeds, users who used the proposed method for a long time are needed. One solution might be to distribute the software of the proposed method. We could implement a function to measure the input speed and evaluate it. However, it would be difficult to have many users of the proposed method. Another solution might be to have a small number of participants who would use the proposed method for a long term. Either way, it would be difficult to measure the performance for one year. Therefore, we think that the immediate solution is to increase the efficiency of the learning of the proposed method. Specifically, better key layouts and learning support software are needed.

8.1.2 Kanji Conversion-required Word Input

The results of the experiments showed that the input speeds of the proposed methods in the kanji conversion-required word input were not very different from those in the sentence input. However, there was a difference between the existing methods. The QWERTY keyboard and the flick keyboard with learning in predictive conversion showed high input speeds. As in the sentence input experiment, a significant difference in the input speeds was shown between the bushu-based layout and the on'yomi-based layout.

Overall, the error rates in the kanji conversion-required word input were lower than those in the sentence input. We think that this is because of the small number of characters entered. However, there was a difference in the error rates between the bushu-based layout and the on'yomi-based layout. We think that the reason is that the participants using the on'yomi-based layout were able to easily associate kanji characters with kana characters, which was not applicable to the bushu-based layout. However, there was still no significant difference between the bushu-based and the on'yomi-based layout. We think that this is because the on'yomi-based layout is not always associative. Since the bushu-based layout is semi-associative and the on'yomi-based layout is associative, the input speeds and the error rates might be reversed depending on the degree of learning. For accurate assessment, we need to find out what is the ratio of the two degrees of learnability.

There was no large difference in the error rates between the on'yomi-based layout and any existing method. According to this result, although it is difficult to compare the input speeds without the long-term use of the methods, we think that the error rates of the methods could be compared even without the long-term use. To reduce the error rate, another associative layout should be proposed and compared with the on'yomi-based layout.

8.1.3 Subjective Evaluation and Participants' Comments

The subjective evaluation showed that the bushu-based layout was of high novelty. However, all the other ratings were low, and in particular, those of perspicuity and efficiency were very low. We think that this is because of the complicated layout that is difficult to use without learning. We also think that this is because the user evaluated the bushu-based and the on'yomi-based layout by comparing them.

The on'yomi-based layout was higher than the bushu-based layout in all the items except novelty. Still, other than stimulation and novelty, it was rated low. One major factor is the short amount of time that the participants spent using the proposed methods. However, in order for users to use it for a long time, the proposed methods should give a good impression at the beginning. The goal is to improve each item to the point that cause no large difference between the proposed methods and the existing methods.

The participants' comments mainly indicated the difficulty of inputting with the proposed methods. The comments included “It is like a mental exercise”, “I’m tired”, and “It is not for me”. The on’yomi-based layout had more positive comments than the bushu-based layout.

8.2 Long-term Experiment

According to the result of the long-term experiment, the input speed of the bushu-based layout after 12 months was about the same as that of the QWERTY keyboard. The long-term experiment was conducted by one of the authors alone, and therefore it is not objective. However, it shows that the input speed of the bushu-based layout could improve with a long-term use.

Based on the author’s experience, we think that there are three stages of growth in the input speeds. The first is to learn the placement of bushu elements. In other words, the user can remember the locations of bushu elements without referring to Figure 5. The next stage is to identify bushu elements of kanji characters whose bushu is confusing. For example, a kanji character is confusing if it has more than one bushu-like element. The final stage is to learn the location of a kanji character. After this, the author knows where frequently used kanji characters are located. We expect that the number of such kanji characters will increase as the author further continues it.

The error rate of the bushu-based layout after 12 months is still higher than that of the QWERTY keyboard. There are several reasons. The first is due to a mistake in the bushu element of a kanji character. Because of the specification of the input, if the user makes a mistake in the bushu, a kana character will be input. The error rate also increases if the user makes a mistake in the kanji character itself. Since some kanji characters have similar shapes, users may make inputting errors. These problems might be solved after a longer-term use.

We have not yet compared the bushu-based layout with other methods with a long-term use. Especially, it would be desirable to include the on’yomi-based layout in the experiment.

9 CONCLUSIONS AND FUTURE WORK

This paper proposed a flick-based tablet software keyboard using direct kanji input. There are two types of layouts: bushu-based and on’yomi-based. The results of the comparative experiments showed that the proposed methods were not as good as the existing methods, even if the hint function was provided. Since the kanji direct input method generally takes long time to learn, one of the authors by himself conducted a long-term experiment. After 12 months, both the input speed and the error rate improved. However, they did not surpass those of the existing method.

Future directions include improving error rates, improving first impressions, evaluating learning efficiency, and exploring more thorough ways to evaluate performance. We aim to improve both the error rate and the first impression by further examining the layout. The evaluation of the input speed requires multiple users who use the proposed methods for a long time. Therefore, by distributing an improved version of the proposed methods, we hope to acquire users who will use them for a long time.

REFERENCES

Apple Inc. (2019). Split and move the keyboard on your iPad. https://support.apple.com/en-us/HT207521.

Bi, X., Chelba, C., Ouyang, T., Partridge, K., and Zhai, S. (2012). Bimanual gesture keyboard. In *Proc. UIST*, pages 137–146.

Fukatsu, Y., Shizuki, B., and Tanaka, J. (2013). No-look flick: Single-handed and eyes-free Japanese text input system on touch screens of mobile devices. In *Proc. MobileHCI*, pages 161–170.

Hakoda, H., Fukatsu, Y., Shizuki, B., and Tanaka, J. (2013). An eyes-free kana input method using two fingers for touch-panel devices. *IPSJ SIG Tech. Rep.*, 154(6):1–8. In Japanese.

Hasegawa, A., Hasegawa, S., and Miyao, M. (2012). Characteristics of the input on software keyboard of tablet devices: Aging effects and differences between the dominant and non-dominant hands for input. *J. Mobile Interact.*, 2(1):23–28.

Ichimura, Y., Saito, Y., Kimura, K., and Hirakawa, H. (2000). Kana-kanji conversion system with input support based on prediction. In *Proc. COLING*, volume 1, pages 341–347.

ISO (1989). Documentation—Romanization of Japanese (kana script). ISO 3602.

Kanev, K., Oido, I., Yoshioka, R., and Mirenkov, N. (2012). Employment of 3D printing for enhanced kanji learning. In *Proc. HCCE*, pages 165–170.

Knox, K. (2012). Designing the Windows 8 touch keyboard. https://docs.microsoft.com/en-us/archive/blogs/b8/designing-the-windows-8-touch-keyboard.

Li, G. and Li, Y. (2019). Chinese Pinyin input method in smartphone era: A literature review study. In *Proc. HCI International*, volume 11569 of *LNCS*, pages 34–43.

Liu, C.-L. and Lin, J.-H. (2008). Using structural information for identifying similar Chinese characters. In *Proc. ACL HLT*, pages 93–96.

Nakamura, Y. and Hosobe, H. (2020). A Japanese bimanual flick keyboard for tablets that improves display space efficiency. In *Proc. VISIGRAPP*, volume 2, pages 170–177.

Niu, J., Zhu, L., Yan, Q., Liu, Y., and Wang, K. (2010). Stroke++: A hybrid Chinese input method for touch screen mobile phones. In *Proc. MobileHCI*, pages 381–382.

Odell, D. (2015). On-screen keyboard: Does the presence of feedback or tactile landmarks improve typing performance? In *Proc. MobileHCI*, pages 131–136.

Plecher, D. A., Eichhorn, C., Kindl, J., Kreisig, S., Wintergerst, M., and Klinker, G. (2018). Dragon tale—a serious game for learning Japanese kanji. In *CHI PLAY Extended Abstracts*, pages 577–583.

Sakurai, Y. and Masui, T. (2013). A flick-based Japanese input system for a QWERTY software keyboard. *IPSJ SIG Tech. Rep.*, 154(5):1–4. In Japanese.

Sax, C., Lau, H., and Lawrence, E. (2011). LiquidKeyboard: An ergonomic, adaptive QWERTY keyboard for touchscreens and surfaces. In *Proc. ICDS*, pages 117–122.

Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017). Construction of a benchmark for the user experience questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4):40–44.

T-Code Project (2003). T-Code laboratory. http://openlab.ring.gr.jp/tcode/index.html. In Japanese.

Takeda, Y. (2019). A survey of the occurrence of kanji characters. In *4th Meeting of the Working Group on the Standard of Japanese Education*. Japanese Agency for Cultural Affairs. In Japanese.

Takei, K. and Hosobe, H. (2018). A 2-by-6 button Japanese software keyboard for tablets. In *Proc. VISIGRAPP*, volume 2, pages 147–154.

Tamaoka, K. (2014). The Japanese writing system and lexical understanding. *Japanese Language and Literature*, 48(2):431–471.

Waragai, H. (2008). *Learning Joyo-Kanji by Reading One Sentence*. Goto Shoin. In Japanese.

Yajima, T. and Hosobe, H. (2018). A Japanese software keyboard for tablets that reduces user fatigue. In *Proc. COMPSAC*, pages 339–346.