



**HAL**  
open science

# SlideKey: Impact of In-depth Previews for a Predictive Text Entry Method

Mathieu Raynal, Benoît Martin

► **To cite this version:**

Mathieu Raynal, Benoît Martin. SlideKey: Impact of In-depth Previews for a Predictive Text Entry Method. 17th International Conference on Computers Helping People with Special Needs - ICCHP 2020, Sep 2020, Lecco, Italy. pp.363-370, 10.1007/978-3-030-58805-2\_43 . hal-02976930

**HAL Id: hal-02976930**

**<https://hal.univ-lorraine.fr/hal-02976930>**

Submitted on 26 Oct 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SlideKey: Impact of In-depth Previews for a Predictive Text Entry Method

Mathieu Raynal<sup>1(✉)</sup> and Benoit Martin<sup>2</sup>

<sup>1</sup> ELIPSE Team, IRIT, University of Toulouse, Toulouse, France  
mathieu.raynal@irit.fr

<sup>2</sup> LCOMS, University of Lorraine, Metz, France  
benoit.martin@univ-lorraine.fr

**Abstract.** Last decade has seen the democratization of small sensitive devices. But text entry solutions remain faithful to the AZERTY or QWERTY layout or to the 12-key mobile phone keypad. We propose SlideKey which is based on FOCL. It uses a linear keyboard whose layout changes according to probabilities. Key selection is operated with the four directional keys. SlideKey integrates a preview of the two-next layouts to ease user to planify his next interactions. Preliminary tests show a not significant gain in performance when the preview is used.

**Keywords:** Slidekey · Soft-keyboard · Character prediction · Preview

## 1 Introduction

With the use of Smartphones, soft keyboards are heavily used. But, before this massive use on Smartphone, soft keyboards already existed on traditional computers. Used with a suitable pointing device, soft keyboards allow people with motor disabilities to replace the physical keyboard. Thus, the soft keyboard makes it possible to enter messages or text on the computer, and consequently, to perform the same tasks as with a physical keyboard.

However, text input using these devices is often very slow. Indeed, these systems being used by means of a cursor or a pointer, it is necessary to move this cursor from one key to another to be able to enter the different characters. Typing text on a physical keyboard can be done with 10 fingers. Therefore, if fingers are well distributed above the keyboard, it is not necessary to move the same finger from one key to another, but to use two separate fingers to enter two successive characters. That's why text input in physical keyboard is much faster.

To compensate for this decrease in input speed, soft keyboards are often coupled with word prediction systems. The most commonly used solutions are to offer the user the most likely words in the form of a list displayed near the soft keyboard [2]. However, these systems are only effective if the desired word is present in the list. In any case, it is often necessary to enter several characters before you can access the word in the list [3].

To improve the input speed, one method is to use a character prediction system. This solution consists of predicting the next character. This makes it easier to point the keys containing the most probable characters either by enlarging these keys [1, 6], by adding additional keys [7], or by increasing the speed of the pointer according to its position [8]. But, this type of solution is mostly used for single-switch scanning using a virtual keyboard [4]. This type of keyboard uses a cursor that switches from one key to another automatically after a delay: for example FOCL [5] or Sybil [10]. When the cursor is on the correct character, the user validates the selection by means of a contactor. After each character entered, the cursor is replaced at the beginning of the keyboard. Therefore, in a single-switch scanning keyboard, without character prediction, if a character is away from the cursor, the user will have to wait for the cursor to move to all keys preceding the desired key. With a character prediction system, the characters are rearranged on the keyboard after each user input, and the most likely characters are positioned closest to the cursor. This limits the number of movements to be performed.

However, the integration of dictionary raises many questions including the cognitive overload induced by the visual inspection of the proposal, the different way to display the proposals to ease navigation, etc. The advantage of such a system resides on a gain higher than the cost that is induced or on a higher user satisfaction. In a broader dialogue context, [11] has shown the benefits of preview. This is important to reduce the cognitive overload caused by the prediction system. Thus, we propose SlideKey a single-switch scanning keyboard that combines a predictive system to a system of preview.

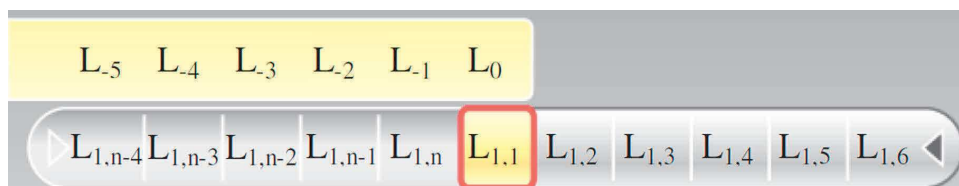
## 2 SlideKey

SlideKey relies on FOCL the keyboard proposed by Scott MacKenzie [5]. Keys are presented on a line in order of their probability of occurrence, taking into account the last or the last two one characters. The selection is done using three keys, two to move from left to right the cursor represented by a blue rectangle, the third for validation.

SlideKey uses the principle of FOCL while distributing characters on a ring. Figure 1 shows its basic layout. The prefix that represents the last six characters entered is displayed in order to prevent the user from unnecessary looks to the input area.  $\forall i \leq 0, L_i$  denotes the last entered letters.  $L_0$  is the last one,  $L_{-1}$  the previous one and so, on.

As FOCL, letters are presented using one line but they are distributed on a ring. The set  $S$  of  $n$  letters actually used contains alphabetical characters and the space character. The ring is partially visible: only eleven letters are displayed to take into account the constraints of the small interactive devices.  $\forall j$  with  $j \in [1, n], L_{1,j}$  denotes one possible letter of the keyboard. The current letter is presented at the middle in the red selection frame ( $L_{1,1}$  in Fig. 1). Unlike FOCL, it is fixed: the characters move (the ring rotates). The prefix and the current letter uses a similar background color to facilitate the reading of the

word being typed. The choice is done with the left and right directions. The down direction validates the current character and the top direction deletes the last entered one.



**Fig. 1.** Basic layout of SlideKey (Color figure online)

We note:

- $P(l, s)$  the probability of character  $l \in S$  to follow the suite of characters  $s = (s_i)_{i \in N}$  with  $s_i \in S - \{\text{space}\}$ .
- $Last_k$  the last  $k$  entered characters. If one or more space characters occur in the last four characters, we consider only the last characters from the last space character.

When validating, the current character is added to the prefix and next possible characters are displayed from left to right from the selection frame. If  $Last_4$  is empty (at the beginning or after a space character), characters are displayed in alphabetical order, the character ‘A’ first and space character at the end. On the contrary, the more probable character is presented in the selection frame and next ones are displayed in order of probability. We have:

$$\sum_{j=1}^n P(L_{1,j}, Last_4) = 1 \quad (1)$$

and

$$\forall j \in [1, n - 1], P(L_{1,j}, Last_4) \leq P(L_{1,j+1}, Last_4) \quad (2)$$

As the ring is closed, we find  $L_{1,n}$  the least common character at the left of the selection frame. In normal use, navigation should mainly use down and right. The left navigation and the display at the left of the selection frame should be only useful at the beginning of a word when characters are displayed in alphabetical order and in case of exceeding the desired character.

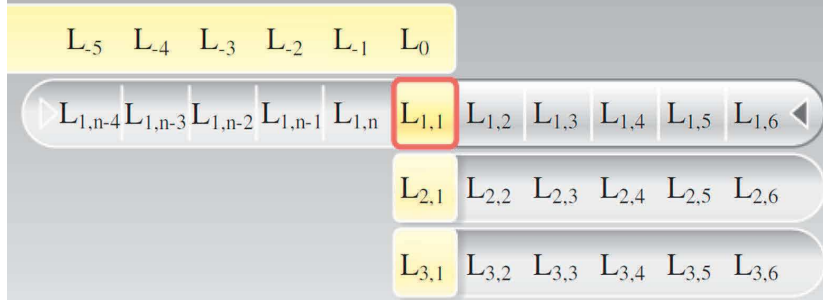
When deleting a character, it is removed from the prefix and the letter most likely to succeed to the corrected prefix is displayed as the new current one.

Figure 2 shows a screenshot of running SlideKey. The user as yet entered the prefix “BONJO” and the system proposes the next characters. The most probable, ‘U’, is displayed in the selection frame. Others, ‘N’, ‘I’, ‘E’, ‘T’, ‘L’, ... ‘M’, ‘W’, ‘X’, ‘Z’, ‘space’, are displayed from left to right in order of probability.

SlideKey is based on the availability of a in-depth preview through two additional lines as shown in Fig. 3.



**Fig. 2.** Running SlideKey (Color figure online)



**Fig. 3.** Basic layout of SlideKey with preview

The second line shows the arrangement in case of validation of the current character  $L_{1,1}$ . We have:

$$\sum_{j=1}^n P(L_{2,j}, Last_3 L_{1,1}) = 1 \quad (3)$$

and

$$\forall j \in [1, n-1], P(L_{2,j}, Last_3 L_{1,1}) \leq P(L_{2,j+1}, Last_3 L_{1,1}) \quad (4)$$

The third line shows the arrangement in case of validation of the most probable character  $L_{2,1}$ . We have:

$$\sum_{j=1}^n P(L_{3,j}, Last_2 L_{1,1} L_{2,1}) = 1 \quad (5)$$

and

$$\forall j \in [1, n-1], P(L_{3,j}, Last_2 L_{1,1} L_{2,1}) \leq P(L_{3,j+1}, Last_2 L_{1,1} L_{2,1}) \quad (6)$$

With this preview,  $W$  the set of suites of characters that are viewable with the in-depth preview is:

$$W = \bigcup_{j \in [1, n]} \{prefix L_{1,1} L_{2,1} L_{3,j}\} \cup \bigcup_{j \in [2, n]} \{prefix L_{1,1} L_{2,j}\} \cup \bigcup_{j \in [3, n]} \{prefix L_{1,j}\} \quad (7)$$

The prefix and the most probable character of each line uses the same background color to ease reading of  $prefix L_{1,1} L_{2,1} L_{3,1}$ , the most probable suite of characters.

Figure 4 shows a screenshot of running SlideKey with the in-depth preview. When the current character is validated the second and third line move up and the third line is replaced by the new prediction. Only one line is new for the user. But if the current character is not the intended one, the second and the third lines are useless: when the current character changes, the first line shifts left or right and the second and the third lines change totally (Fig. 4).



**Fig. 4.** Running SlideKey with the in-depth preview (Color figure online)

Generally, with the dynamic layout the cognitive load is important. The preview should allow anticipating the interaction to be performed to achieve the desired characters but it can involve a visual and cognitive overload too. It is difficult to quantify the influence of these parameters on the performance and the user satisfaction without conducting an experiment in real situation.

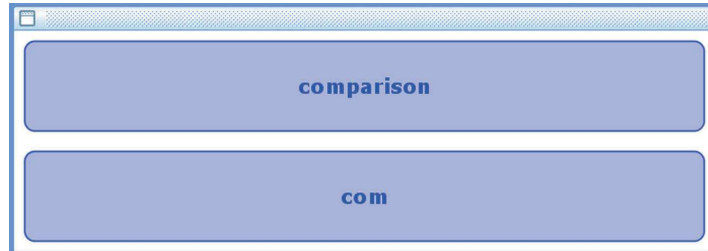
### 3 Experimentation

**Participants** - Height participants, two female and six male, participated in the experiment. They ranged in age from 22 to 40. There were seven right-handed and all are computer specialists.

**Apparatus** - The experiment was conducted on a Dell laptop running on windows. A joystick was attached to the system and was the only device used by participants. We chose for this first experimentation to restrict us to a soft keyboard which contained only the 26 characters of the Latin alphabet and the space character. Moving on the keyboard was made using the directional arrows of a joystick. Experimental software including the keyboard and the prediction system was done in Java. Data collection was performed with the platform E-Assist [9]. All data were saved in an XML file.

**Design** - Each subject had to perform two exercises of copy with SlideKey: one with the preview and one without preview. For each exercise, the subjects had to copy out 22 words which were the same for the two exercises. Words were chosen as the most usually used in the maternal language of the users and as to represent a maximum of different co-occurrences. Before each exercise, subjects have a training session where they had to type 12 words with the system proposed. We used a counterbalanced, within-subjects design.

**Procedure** - The word to be copied was presented on a line, and the word being typed by the user appeared on the line below (Fig. 5). The text entry errors were not displayed on the screen. Instead there was a visual and audio feedback signaling the error and the strip did not move until the subject entered the right character. At the end of each word, the participant had to enter the space character.



**Fig. 5.** Experimental device

## 4 Results and Discussion

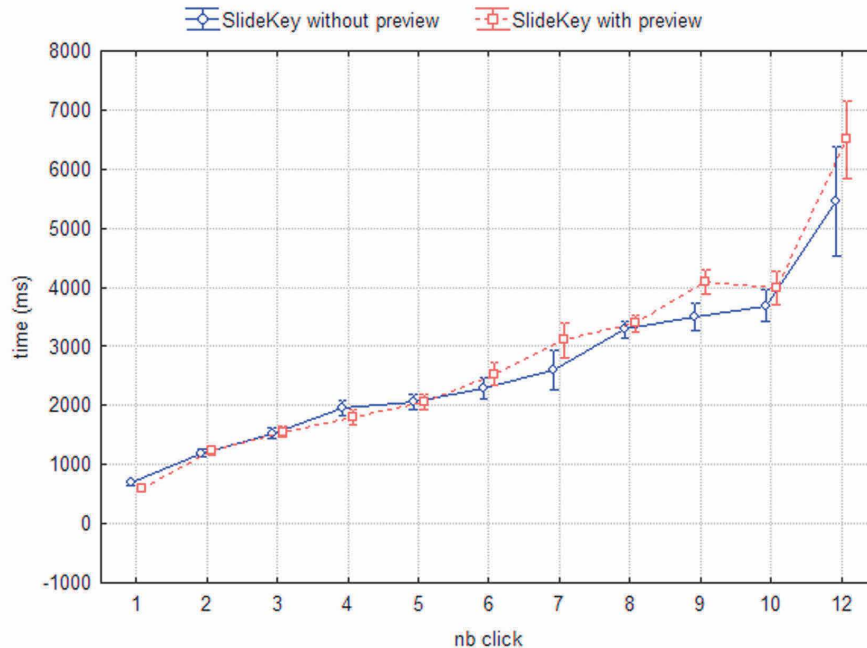
Table 1 shows the mean time that participants have to type a word ( $MT_{word}$ ), and more precisely the mean time to enter a character when this one is the most likely ( $MT_{char}$ ). On average, participants entered a word in 7756 ms without the preview, whereas they needed only 7534 ms with the preview. Thus they gain on average 3% of time with the preview relatively to SlideKey without preview. However, these results remain to be confirmed because the statistical test performed (ANOVA) showed no significant differences between the two systems ( $F = 0.59418$  with  $p = 0.44133$ ).

**Table 1.** Mean time (in ms) to input a word ( $MT_{word}$ ) and a character when it is the most likely ( $MT_{char}$ ).

Heading level	$MT_{word}$	$MT_{char}$
With preview	7.534	585
Without preview	7.756	691

However, a closer analysis shows significant differences between the two methods tested. The first significant difference is the input of a character which is the best result of the prediction system. Indeed, after the validation of a character, the most likely character appears in the selection frame (red outline on Fig. 6). With preview, before entering the current character, we can see which character has the best chance of succeeding him in the yellow area just below the selection frame. In this case, after entering the current character, the next

character is entered just by another tap on the down arrow of the joystick. We see a significant difference ( $F = 42.164$ ,  $p < 0.0001$ ) between the time required to input this character with SlideKey with preview and without preview (cf. Table 1). On average, participants entered this character in 691 ms without the preview, whereas they needed only 585 ms with the preview (that is to say that text input time decrease on average 15).



**Fig. 6.** Mean time to enter a character according to its position on the line

Finally, we see in Fig. 6 that, depending on the character position on the line, the system with preview is useful or not. The limit of the efficiency of preview is the 5<sup>th</sup> character. This shows that the system of preview is useful when the user can view the position of the next character. However, if the next character to enter is not previewed (because too far down the prediction list), the preview becomes more disruptive to the user.

## 5 Conclusion

We presented SlideKey, a soft keyboard offering a preview of the two next character layouts to help the user with the following interactions. Preliminary tests show that preview offers a significant gain when the most likely character indicated by the prediction is the character to enter. However, on all typed characters, even if a gain is observed, it is too small to be significant.

These results confirm our proposal for these preliminary tests. The prediction module was reset for each word. Thus, the module n-gram was “underutilized” in early word for lack of character. Hopefully better prediction results with phrases



and thus a significant gain. In addition, results show that the preview is useful if the next character to enter is previewed. Then, we plan to increase the number of characters at the right of the current one by reducing the space at its left. These works will be tested later.

## References

1. Aulagner, G., François, R., Martin, B., Michel, D., Raynal, M.: Floodkey: increasing software keyboard keys by reducing needless ones without occultation. In: Proceedings of the 10th WSEAS International Conference on Applied Computer Science, pp. 412–417. World Scientific and Engineering Academy and Society (WSEAS) (2010)
2. Badr, G., Raynal, M.: WordTree: results of a word prediction system presented thanks to a tree. In: Stephanidis, C. (ed.) UAHCI 2009. LNCS, vol. 5616, pp. 463–471. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-02713-0\\_49](https://doi.org/10.1007/978-3-642-02713-0_49)
3. Badr, G., Raynal, M.: Evaluation of WordTree system with motor disabled users. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2010. LNCS, vol. 6180, pp. 104–111. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14100-3\\_17](https://doi.org/10.1007/978-3-642-14100-3_17)
4. MacKenzie, I.S.: Modeling text input for single-switch scanning. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) ICCHP 2012. LNCS, vol. 7383, pp. 423–430. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-31534-3\\_63](https://doi.org/10.1007/978-3-642-31534-3_63)
5. MacKenzie, S.: Mobile text entry using three keys. In: Proceedings of the second Nordic Conference on Human-Computer Interaction, pp. 27–34. ACM (2002)
6. Merlin, B., Raynal, M.: Evaluation of SpreadKey system with motor impaired users. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2010. LNCS, vol. 6180, pp. 112–119. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14100-3\\_18](https://doi.org/10.1007/978-3-642-14100-3_18)
7. Raynal, M.: KeyGlasses: semi-transparent keys on soft keyboard. In: Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility, pp. 347–349 (2014)
8. Raynal, M., MacKenzie, I.S., Merlin, B.: Semantic keyboard: fast movements between keys of a soft keyboard. In: Miesenberger, K., Fels, D., Archambault, D., Peñáz, P., Zagler, W. (eds.) ICCHP 2014. LNCS, vol. 8548, pp. 195–202. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-08599-9\\_30](https://doi.org/10.1007/978-3-319-08599-9_30)
9. Raynal, M., Maubert, S., Vigouroux, N., Vella, F., Magnien, L.: E-assiste: a platform allowing evaluation of text input systems. In: 3rd International Conference on Universal Access in Human-Computer Interaction (UAHCI 2005), Las Vegas, USA (2005). <https://hal.archives-ouvertes.fr/hal-01761479>
10. Schadle, I.: Sibyl: AAC system using NLP techniques. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 1009–1015. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-27817-7\\_149](https://doi.org/10.1007/978-3-540-27817-7_149)
11. Sellen, A.J., Kurtenbach, G.P., Buxton, W.A.: The prevention of mode errors through sensory feedback. *Hum.-Comput. Interact.* **7**(2), 141–164 (1992)